GI-Edition

Lecture Notes in Informatics

Norbert Ritter, Andreas Henrich,
Wolfgang Lehner, Andreas Thor,
Steffen Friedrich, Wolfram Wingerath
(Hrsg.)

Datenbanksysteme für Business, Technologie und Web (BTW 2015) – Workshopband

02. – 03. März 2015
Hamburg

Proceedings

242

Norbert Ritter, Andreas Henrich, Wolfgang Lehner, Andreas Thor,
Steffen Friedrich, Wolfram Wingerath (Hrsg.): BTW 2015 – Workshopband

Norbert Ritter, Andreas Henrich,
Wolfgang Lehner, Andreas Thor, Steffen Friedrich,
Wolfram Wingerath (Hrsg.)

Datenbanksysteme für Business, Technologie und Web (BTW 2015)

Workshopband

02. – 03.03.2015
in Hamburg, Germany

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-242

ISBN 978-3-88579-636-7
ISSN 1617-5468

Volume Editors

Norbert Ritter
Universität Hamburg
Fachbereich Informatik
Datenbanken und Informationssysteme
22527 Hamburg, Germany
E-Mail: ritter@informatik.uni-hamburg.de

Andreas Henrich
Otto-Friedrich-Universität Bamberg
Fakultät Wirtschaftsinformatik und Angewandte Informatik
Lehrstuhl für Medieninformatik
96047 Bamberg, Germany
E-Mail: andreas.henrich@uni-bamberg.de

Wolfgang Lehner
Technische Universität Dresden
Fakultät Informatik
Institut für Systemarchitektur
01062 Dresden, Germany
Email: wolfgang.lehner@tu-dresden.de

Andreas Thor
Deutsche Telekom Hochschule für Telekommunikation Leipzig
Gustav-Freytag-Str. 43-45
04277 Leipzig, Germany
E-Mail: thor@hft-leipzig.de

Steffen Friedrich
Universität Hamburg
Fachbereich Informatik
Datenbanken und Informationssysteme
22527 Hamburg, Germany
E-Mail: friedrich@informatik.uni-hamburg.de

Wolfram Wingerath
Universität Hamburg
Fachbereich Informatik
Datenbanken und Informationssysteme
22527 Hamburg, Germany
E-Mail: wingerath@informatik.uni-hamburg.de

Series Editorial Board
Heinrich C. Mayr, Alpen-Adria-Universität Klagenfurt, Austria
(Chairman, mayr@ifit.uni-klu.ac.at)
Dieter Fellner, Technische Universität Darmstadt, Germany
Ulrich Flegel, Hochschule für Technik, Stuttgart, Germany
Ulrich Frank, Universität Duisburg-Essen, Germany
Johann-Christoph Freytag, Humboldt-Universität zu Berlin, Germany
Michael Goedicke, Universität Duisburg-Essen, Germany
Ralf Hofestädt, Universität Bielefeld, Germany
Michael Koch, Universität der Bundeswehr München, Germany
Axel Lehmann, Universität der Bundeswehr München, Germany
Peter Sanders, Karlsruher Institut für Technologie (KIT), Germany
Sigrid Schubert, Universität Siegen, Germany
Ingo Timm, Universität Trier, Germany
Karin Vosseberg, Hochschule Bremerhaven, Germany
Maria Wimmer, Universität Koblenz-Landau, Germany

Dissertations
Steffen Hölldobler, Technische Universität Dresden, Germany
Seminars
Reinhard Wilhelm, Universität des Saarlandes, Germany
Thematics
Andreas Oberweis, Karlsruher Institut für Technologie (KIT), Germany

© Gesellschaft für Informatik, Bonn 2015
printed by Köllen Druck+Verlag GmbH, Bonn

Vorwort

In den letzten Jahren hat es auf dem Gebiet des Datenmanagements große Veränderungen gegeben. Dabei muss sich die Datenbankforschungsgemeinschaft insbesondere den Herausforderungen von „Big Data“ stellen, welche die Analyse von riesigen Datenmengen unterschiedlicher Struktur mit kurzen Antwortzeiten im Fokus haben. Neben klassisch strukturierten Daten müssen moderne Datenbanksysteme und Anwendungen semistrukturierte, textuelle und andere multimodale Daten sowie Datenströme in völlig neuen Größenordnungen verwalten. Gleichzeitig müssen die Verarbeitungssysteme die Korrektheit und Konsistenz der Daten sicherstellen.

Die jüngsten Fortschritte bei Hardware und Rechnerarchitektur ermöglichen neuartige Datenmanagementtechniken, die von neuen Index- und Anfrageverarbeitungsparadigmen (In-Memory, SIMD, Multicore) bis zu neuartigen Speichertechniken (Flash, Remote Memory) reichen. Diese Entwicklungen spiegeln sich in aktuell relevanten Themen wie Informationsextraktion, Informationsintegration, Data Analytics, Web Data Management, Service-Oriented Architectures, Cloud Computing oder Virtualisierung wider.

Wie auf jeder BTW-Konferenz gruppieren sich um die Tagung eine Reihe von Workshops, die spezielle Themen in kleinen Gruppen aufgreifen und diskutieren. Im Rahmen der BTW 2015 finden folgende Workshops statt:

- Databases in Biometrics, Forensics and Security Applications: DBforBFS
- Data Streams and Event Processing: DSEP
- Data Management for Science: DMS

Dabei fasst der letztgenannte Workshop DMS als Joint Workshop die beiden Initiativen Big Data in Science (BigDS) und Data Management for Life Sciences (DMforLS) zusammen.

Mit seinen Schwerpunkten reflektiert das Workshopprogramm aktuelle Forschungsgebiete von hoher praktischer Relevanz. Zusätzlich präsentieren Studenten im Rahmen des Studierendenprogramms die Ergebnisse ihrer aktuellen Abschlussarbeiten im Bereich Datenmanagement. Für jeden Geschmack sollte sich somit ein Betätigungsfeld finden lassen!

Die Materialien zur BTW 2015 werden auch über die Tagung hinaus unter http://www.btw-2015.de zur Verfügung stehen.

Die Organisation einer so großen Tagung wie der BTW mit ihren angeschlossenen Veranstaltungen ist nicht ohne zahlreiche Partner und Unterstützer möglich. Sie sind auf den folgenden Seiten aufgeführt. Ihnen gilt unser besonderer Dank ebenso wie den Sponsoren der Tagung und der GI-Geschäftsstelle.

Hamburg, Bamberg, Dresden, Leipzig, im Januar 2015

Norbert Ritter, Tagungsleitung und Vorsitzender des Organisationskomitees

Andreas Henrich und Wolfgang Lehner, Leitung Workshopkomitee

Andreas Thor, Leitung Studierendenprogramm

Wolfram Wingerath, Steffen Friedrich, Tagungsband und Organisationskomitee

Tagungsleitung
Norbert Ritter, Universität Hamburg

Organisationskomitee
Norbert Ritter
Anne Hansen-Awizen
Steffen Friedrich
Wolfram Wingerath
Felix Gessert
Fabian Panse
Volker Nötzold

Studierendenprogramm
Andreas Thor, HfT Leipzig

Koordination Workshops
Andreas Henrich, Univ. Bamberg
Wolfgang Lehner, TU Dresden

Tutorienprogramm
Norbert Ritter, Univ. Hamburg
Thomas Seidl, RWTH Aachen
Andreas Henrich, Univ. Bamberg
Wolfgang Lehner, TU Dresden

Second Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS)
Vorsitz: Jana Dittmann, Univ. Magdeburg; Veit Köppen, Univ. Magdeburg; Gunter Saake, Univ. Magdeburg; Claus Vielhauer, FH Brandenburg

Ruediger Grimm, Univ. Koblenz
Dominic Heutelbeck, FTK
Stefan Katzenbeisser, TU Darmstadt
Claus-Peter Klas, GESIS
Günther Pernul, Univ. Regensburg
Ingo Schmitt, BTU Cottbus
Claus Vielhauer, FH Brandenburg
Sviatoslav Voloshynovskiy, UNIGE, CH
Edgar R. Weippl, SBA Research, Austria

Data Streams and Event Processing (DSEP)
Vorsitz: Marco Grawunder, Univ. Oldenburg, Daniela Nicklas Univ. Bamberg

Andreas Behrend, Univ. Bonn
Klemens Boehm, KIT
Peter Fischer, Univ. Freiburg
Dieter Gawlick, Oracle
Boris Koldehofe, TU Darmstadt
Wolfgang Lehner, TU Dresden
Richard Lenz, Univ. Erlangen-Nürnberg
Klaus Meyer-Wegener, Univ. Erlangen-Nürnberg
Gero Mühl, Univ. Rostock
Kai-Uwe Sattler, TU Ilmenau
Thorsten Schöler, HS Augsburg

Joint Workshop on Data Management for Science (DMS)

Workshop on Big Data in Science (BigDS)

Vorsitz: Birgitta König-Ries, Univ. Jena; Erhard Rahm, Univ. Leipzig; Bernhard Seeger, Univ. Marburg

Alsayed Algergawy, Univ. Jena
Peter Baumann, Jacobs Univ.
Matthias Bräger, CERN
Thomas Brinkhoff, FH Oldenburg
Michael Diepenbroeck, AWI
Christoph Freytag, HU Berlin
Michael Gertz, Univ. Heidelberg
Frank-Oliver Glöckner, MPI-MM
Anton Güntsch, BGBM Berlin-Dahlem
Thomas Heinis, IC, London
Thomas Hickler, Senckenberg
Jens Kattge, MPI für Biogeochemie
Alfons Kemper, TU München
Meike Klettke, Univ. Rostock
Alex Markowetz, Univ. Bonn
Thomas Nauss, Univ. Marburg
Jens Nieschulze, Univ. Göttingen
Kai-Uwe Sattler, TU Ilmenau
Stefanie Scherzinger, OTH Regensburg
Myro Spiliopoulou, Univ. Magdeburg
Uta Störl, Hochschule Darmstadt

Workshop on Data Management for Life Sciences (DMforLS)

Vorsitz: Sebastian Dorok, Bayer Pharma AG; Matthias Lange, IPK Gatersleben; Gunter Saake, Univ. Magdeburg

Sebastian Breß, TU Dortmund
Sebastian Dorok, Bayer Pharma AG
Mourad Elloumi, UTM Tunisia
Ralf Hofestädt, Univ. Bielefeld
Andreas Keller, Saarland Univ.
Jacob Köhler, DOW AgroSciences
Horstfried Läpple, Bayer HealthCare
Matthias Lange, IPK Gatersleben
Ulf Leser, HU Berlin
Wolfgang Müller, HITS GmbH
Erhard Rahm, Univ. Leipzig
Gunter Saake, Univ. Magdeburg
Uwe Scholz, IPK Gatersleben
Can Türker, ETH Zürich

Inhaltsverzeichnis

Workshopprogramm

Second Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS)

Jana Dittmann, Veit Köppen, Gunter Saake, Claus Vielhauer
Second Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS) 19

Veit Köppen, Mario Hildebrandt, Martin Schäler
On Performance Optimization Potentials Regarding Data Classification in Forensics 21

Maik Schott, Claus Vielhauer, Christian Krätzer
Using Different Encryption Schemes for Secure Deletion While Supporting Queries 37

Data Streams and Event Processing (DSEP)

Marco Grawunder, Daniela Nicklas
Data Streams and Event Processing (DSEP) 49

Timo Michelsen, Michael Brand, H.-Jürgen Appelrath
Modulares Verteilungskonzept für Datenstrommanagementsysteme 51

Niko Pollner, Christian Steudtner, Klaus Meyer-Wegener
Placement-Safe Operator-Graph Changes in Distributed Heterogeneous Data Stream Systems 61

Michael Brand, Tobias Brandt, Carsten Cordes, Marc Wilken, Timo Michelsen
Herakles: A System for Sensor-Based Live Sport Analysis using Private Peer-to-Peer Networks 71

Christian Kuka, Daniela Nicklas
Bestimmung von Datenunsicherheit in einem probabilistischen Datenstrommanagementsystem 81

Cornelius A. Ludmann, Marco Grawunder, Timo Michelsen, H.-Jürgen Appelrath
Kontinuierliche Evaluation von kollaborativen Recommender-Systeme in Datenstrommanagementsystemen 91

Sebastian Herbst, Johannes Tenschert, Klaus Meyer-Wegener
Using Data-Stream and Complex-Event Processing to Identify Activities of Bats 93

Peter M. Fischer, Io Taxidou
Streaming Analysis of Information Diffusion 95

Henrik Surm, Daniela Nicklas
Towards a Framework for Sensor-based Research and Development Platform for Critical, Socio-technical Systems 97

Felix Beier, Kai-Uwe Sattler, Christoph Dinh, Daniel Baumgarten
Dataflow Programming for Big Engineering Data 101

Joint Workshop on Data Management for Science (DMS)

Sebastian Dorok, Birgitta König-Ries, Matthias Lange, Erhard Rahm, Gunter Saake, Bernhard Seeger
Joint Workshop on Data Management for Science (DMS) 105

Alexandr Uciteli, Toralf Kirsten
Ontology-based Retrieval of Scientific Data in LIFE 109

Christian Colmsee, Jinbo Chen, Kerstin Schneider, Uwe Scholz, Matthias Lange
Improving Search Results in Life Science by Recommendations based on Semantic Information 115

Marc Schäfer, Johannes Schildgen, Stefan Deßloch
Sampling with Incremental MapReduce 121

Andreas Heuer
METIS in PArADISE Provenance Management bei der Auswertung von Sensordatenmengen für die Entwicklung von Assistenzsystemen 131

Martin Scharm, Dagmar Waltemath
Extracting reproducible simulation studies from model repositories using the CombineArchive Toolkit 137

Robin Cijvat, Stefan Manegold, Martin Kersten, Gunnar W. Klau, Alexander Schönhuth, Tobias Marschall, Ying Zhang
Genome sequence analysis with MonetDB: a case study on Ebola virus diversity 143

Ahmet Bulut
RightInsight: Open Source Architecture for Data Science 151

Christian Authmann, Christian Beilschmidt, Johannes Drönner, Michael Mattig, Bernhard Seeger
Rethinking Spatial Processing in Data-Intensive Science 161

Studierendenprogramm

Marc Büngener
CBIR gestütztes Gemälde-Browsing 173

David Englmeier, Nina Hubig, Sebastian Goebl, Christian Böhm
Musical Similarity Analysis based on Chroma Features and Text Retrieval Methods 183

Alexander Askinadze
Vergleich von Distanzen und Kernel für Klassifikatoren zur Optimierung der Annotation von Bildern 193

Matthias Liebeck
Aspekte einer automatischen Meinungsbildungsanalyse von Online-Diskussionen 203

Martin Winter, Sebastian Goebl, Nina Hubig, Christopher Pleines, Christian Böhm
Development and Evaluation of a Facebook-based Product Advisor for Online Dating Sites 213

Daniel Töws, Marwan Hassani, Christian Beecks, Thomas Seidl
Optimizing Sequential Pattern Mining Within Multiple Streams 223

Marcus Pinnecke
Konzept und prototypische Implementierung eines föderativen Complex Event Processing Systeme mit Operatorverteilung 233

Monika Walter, Axel Hahn
Unterstützung von datengetriebenen Prozessschritten in Simulationsstudien durch Verwendung multidimensionaler Datenmodelle 243

Niklas Wilcke
DduP – Towards a Deduplication Framework utilising Apache Spark 253

Tutorienprogramm

Christian Beecks, Merih Uysal, Thomas Seidl
Distance-based Multimedia Indexing 265

Kai-Uwe Sattler, Jens Teubner, Felix Beier, Sebastian Breß
Many-Core-Architekturen zur Datenbankbeschleunigung 269

Felix Gessert, Norbert Ritter
Skalierbare NoSQL- und Cloud-Datenbanken in Forschung und Praxis 271

Jens Albrecht, Uta Störl
Big-Data-Anwendungsentwicklung mit SQL und NoSQL 275

Workshopprogramm

Second Workshop on Databases in Biometrics, Forensics and Security Applications

Second Workshop on Databases in Biometrics, Forensics and Security Applications

Jana Dittmann[1], jana.dittmann@ovgu.de
Veit Köppen[1], veit.koeppen@ovgu.de
Gunter Saake[1], gunter.saake@ovgu.de
Claus Vielhauer[2], claus.vielhauer@fh-brandenburg.de

[1]Otto-von-Guericke-University Magdeburg
[2]Brandenburg University of Applied Science

The 1st Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS) was held as satellite workshop of the BTW 2013. The workshop series is intended for disseminating knowledge in the areas of databases in the focus for biometrics, forensics, and security complementing the regular conference program by providing a place for in-depth discussions of this specialized topic. The workshop will consist of two parts: First, presentation of accepted workshop papers and second, a discussion round. In the discussion round, the participants will derive research questions and goals to address important issues in the domain databases and security. We expect the workshop to facilitate cross-fertilization of ideas among key stakeholders from academia, industry, practitioners and government agencies. Theoretical and practical coverage of the topics will be considered. We also welcome software and hardware demos. Full and short papers are solicited.

Motivated by today's challenges from both disciplines several topics include but are not limited to:

- approaches increasing the search speed in databases for biometrics, forensics and security,
- database validation procedures for integrity verification of digital stored content
- design aspects to support multimodal biometric evidence and its combination with other forensic evidence
- interoperability methodologies and exchange protocols of data of large-scale operational (multimodal) databases of identities and biometric data for forensic case assessment and interpretation, forensic intelligence and forensic ID management
- database security evaluation and benchmarks for forensics and biometric applications
- the role of databases in emerging applications in Biometrics and Forensics
- privacy, policy, legal issues, and technologies in databases of biometric, forensic and security data.

1 Workshop Organizers

Jana Dittmann (Otto-von-Guericke-University Magdeburg)
Veit Köppen (Otto-von-Guericke-University Magdeburg)
Gunter Saake (Otto von Guericke University Magdeburg)
Claus Vielhauer (Brandenburg University of Applied Science)

2 Program Committee

Ruediger Grimm (University of Koblenz, DE)
Dominic Heutelbeck (FTK, DE)
Stefan Katzenbeisser (Technical University Darmstadt, DE)
Claus-Peter Klas (GESIS, DE)
Günther Pernul (Universität Regensburg, DE)
Ingo Schmitt (Brandenburg University of Technology, DE)
Claus Vielhauer (Brandenburg University of Applied Science, DE)
Sviatoslav Voloshynovskiy (unige, CH)
Edgar R. Weippl (sba-research, Austria)

On Performance Optimization Potentials Regarding Data Classification in Forensics

Veit Köppen, Mario Hildebrandt, Martin Schäler

Faculty of Computer Science
Otto-von-Guericke-University Magdeburg
Universitätsplatz 2
39106 Magdeburg
veit.koeppen@ovgu.de
mario.hildbrandt@ovgu.de
martin.schaeler@ovgu.de

Abstract: Classification of given data sets according to a training set is one of the essentials bread and butter tools in machine learning. There are several application scenarios, reaching from the detection of spam and non-spam mails to recognition of malicious behavior, or other forensic use cases. To this end, there are several approaches that can be used to train such classifiers. Often, scientists use machine learning suites, such as WEKA, ELKI, or RapidMiner in order to try different classifiers that deliver best results. The basic purpose of these suites is their easy application and extension with new approaches. This, however, results in the property that the implementation of the classifier is and cannot be optimized with respect to response time. This is due to the different focus of these suites. However, we argue that especially in basic research, systematic testing of different promising approaches is the default approach. Thus, optimization for response time should be taken into consideration as well, especially for large scale data sets as they are common for forensic use cases. To this end, we discuss in this paper, in how far well-known approaches from databases can be applied and in how far they affect the classification result of a real-world forensic use case. The results of our analyses are points and respective approaches where such performance optimizations are most promising. As a first step, we evaluate computation times and model quality in a case study on separating latent fingerprint patterns.

1 Motivation

Data are drastically increased in a given time period. This is not only true for the number of data sets (comparable to new data entries), but also with respect to dimensionality. To get into control of this information overload, data mining techniques are used to identify patterns within the data. Different application domains require for similar techniques and therefore, can be improved as the general method is enhanced.

In our application scenario, we are interested in the identification of patterns in data that are acquired from latent fingerprints. Within the acquired scanned data a two-class classification is of interest, to identify the fingerprint trace and the background noise. As point

of origin, experts classify different cases. This supervised approach is used to learn a classification and thus, to support experts in their daily work. With a small number of scanned data sets that the expert has to check and classify, a high number of further data sets can be automatically classified.

Currently, the system works in a semi-automatic process and several manual steps have to be performed. Within this paper, we investigate the influence on system response and model quality, in terms of accuracy and precision, in the context of integrating the data and corresponding processes in a holistic system. Although a complete integration is feasible, different tools are currently used, which do not fully cooperate. Therefore, the efficiency or optimization regarding computation or response time are not in the focus of this work. With this paper, we step forward to create a cooperating and integrated environment that performs efficient with respect to model quality.

This paper is structured as follows: In the next section, we briefly present some background regarding classification and database technologies for accessing multi-dimensional data. In Section 3, we describe the case study that is the motivation for our analysis. Within Section 4, we present our evaluation on the case study data regarding optimization due to feature and data space reduction. Finally, we conclude our work in Section 5.

2 Background

In this section, we give background on classification algorithms in general. Then, we explain one of these algorithms that we apply in the remainder of this paper in more details. Finally, we introduce promising optimization approaches known from databases. We use these approaches in the remainder to discuss their optimization potential with respect to classification.

2.1 Classification Algorithms

In the context of our case study in Section 3, several classification algorithms can be utilized, see, e.g., [MKH+13]. Each of those algorithms is used for supervised learning. Such type of learning consists of a model generation based on training data, which are labeled according to a ground-truth. The utilized classification algorithms in [MKH+13] partition the feature space to resemble the distribution of each instance (data point) in this space. Afterward, the quality of the model can be evaluated using an independent set of labeled test data by comparing the decision of the classifier with the assigned label.

The utilized classification schemes from the WEKA data mining software [HFH+09] in [MKH+13] include support vector machines, multilayer perceptrons, rule based classifiers, decision trees, and ensemble classifiers. The latter ones combine multiple models in their decision process.

C4.5 decision tree

In this paper, we use the classifier J48, WEKA's [HFH+09] implementation of the fast C4.5 decision tree [Qui93], which is an improvement of the ID 3 algorithm [Qui86] and one of the most widely known decision tree classifiers for such problems. The advantage of decision trees is their comprehensiveness: the classifier's decision is a leaf reached by a path of single feature thresholds. The size of the tree is reduced by a pruning algorithm which replaces subtrees. Furthermore, this particular implementation is able to deal with missing values. In order to do that, the distribution of the available values for this particular feature is taken into account.

```
 1 build_tree ( Data (R:{r_1,..,,r_n},C)
 2     R: non-categorical_attributes r_1 to r_n,
 3     C: categorical attribute,
 4     S: training set in same schema as Data)
 5 returning decision_tree;
 6
 7 begin
 8     -- begin exceptions
 9     If is_empty(S)
10         return FAILURE;
11     If only_one_category(DATA)
12         return single_node_tree(value(C));
13     If is_empty(R)
14         return single_node_tree(most_frequent_value(C));
15     -- end excpetions
16     Attribute r_d (elem of R) := largest_Gain(R,S);
17     {d_i| i=1,2, .., m} := values_of_attribute(r_d);
18     {S_i| i=1,2, .., k} := subsets(S) where in each subset value(r_d) = d_i holds;
19     decision_tree := tree(r_d) with nodes { d_1, d_2, .., d_m} pointing to trees
20     call ID3((R-{r_d}, C), S1), ID3((R-{r_d}), C, S2), .., ID3((R-{r_d}, C), S_k);
21     return decision_tree;
22 end build_tree;
```

Figure 1: Algorithm to build a C4.5 decision tree, adapted from [Qui86]

In Figure 1, we depict the general algorithm to build a C4.5 decision tree. The argument for the algorithm is a training set consisting of: (1) n non-categorical attributes R reaching from r_1 to r_n, (2) the categorical attribute (e.g., spam or not spam), and (3) a training set with the same schema. In Lines 8 to 15, the exception handling is depicted, for instance if there are only spam mails (Line 11). The actual algorithm tries to find the best attribute r_d and distributes the remaining tuples in S according to their value in r_d. For each subtree that is created in that way the algorithm is called recursively.

2.2 Approaches for Efficient Data Access

Data within databases have to be organized in such a way that they are efficiently accessed. In the case of multi-dimensional data, an intuitive order does not exist. This is even more apparent for the identification of unknown patterns, where an ordering in a multi-dimensional space always dominates some dimensions. For these reasons, differ-

ent approaches have been proposed. They can be differentiated into storage and index structures.

Typical storage improvements within the domain of Data Warehousing [KSS14a] are column-oriented storage [AMH08], Iceberg-Cube [FSGM+98], and Data Dwarf [SDRK02]. Whereas the Iceberg-Cube reduces computational effort, column-oriented storage improves the I/O with respect to the application scenario, where operations are performed in a column-oriented way. The Data Dwarf heavily reduces the stored data volume without loss of information. It combines computational effort and I/O cost for improving efficiency.

Furthermore, there exist many different index structures for specialized purposes [GBS+12]. Very well-known index structures for multi-dimensional purposes are the kd-Tree [Ben75] and R-Tree [Gut84]. Both mentioned indexes are candidates, which suffer especially from the curse of dimensionality. The curse of dimensionality is a property of large and sparsely populated high-dimensional spaces, which results in the effect that for tree-based indexes often large parts have to be taken into consideration for a query (e.g., because of node overlaps). To this end, several index structures, as the Pyramid technique [BBK98] or improved sequential scans, such as the VA-File [WB97] are proposed. In the following, we briefly explain some well-known indexes that, according to prior evaluations [SGS+13], result in a significant performance increase. A broader overview on index structures can be found in [GG98] or [BBK01]. Challenges regarding parameterization of index structures as well as implementation issues are discussed in [AKZ08, KSS14b, SGS+13].

2.2.1 Column vs. Row Stores

Traditionally, database systems store their data row-wise. That means that each tuple with all its attributes is stored and then the next tuple follows. By contrast, columnar storage means that all values of a column are stored sequentially and then the next column follows. Dependent on the access pattern of a classification algorithm, the traditional row-based storage should be replaced if, for instance, one dimension (column) is analyzed to find an optimal split in this dimension. In this case, we expect a significant performance benefit.

2.2.2 Data Dwarf

The basic idea of the Data Dwarf storage structure is to use prefix and suffix redundancies for multi-dimensional points to compress the data. For instance, the three dimensional points $A(1, 2, 3)$ and $B(1, 2, 4)$ share the same pre-fix $(1, 2, _)$. As a result, the Dwarf has two interesting effects that are able to speed-up classifications. Firstly, due to the compression, we achieve an increased caching performance. Secondly, the access path is stable, which means that we require exactly the number of dimension look-ups to find a point (e.g., three look-ups for three dimensional points).

2.2.3 kd-Tree

A kd-Tree index is a multi-dimensional adaption of the well-known B-Tree cycling through the available dimensions. Per tree level, this index distributes the remaining points in the current subtree into two groups. One group in the left subtree where the points have a value smaller or equal than the separator value in the current dimension, while the remaining points belong to the right sub tree. The basic goal is to achieve logarithmic effort for exact match queries. In summary, this index structure can be used to efficiently access and analyze single dimensions in order to separate two classes.

2.2.4 VA-File

Tree-based index structures suffer from the curse of dimensionality. This may result in the effect that they are slower than a sequential scan. To this end, improvements of the sequential scan are proposed. The basic idea of the Vector Approximation File is to use a compressed *approximation* of the existing data set that fits into the main memory (or caches). On this compressed data an initial filter step is performed in order to minimize actual point look-ups. In how far this technique can be applied to speed-up classifications is currently unknown.

3 Case Study

As described in [HKDV14], the classification of contact-less scans of latent fingerprints is performed using a block based approach. The following subsections summarize the application scenario, the data gathering process, and a description of the feature space. We depict this process in Fig. 2. We describe the steps in the following in more detail.

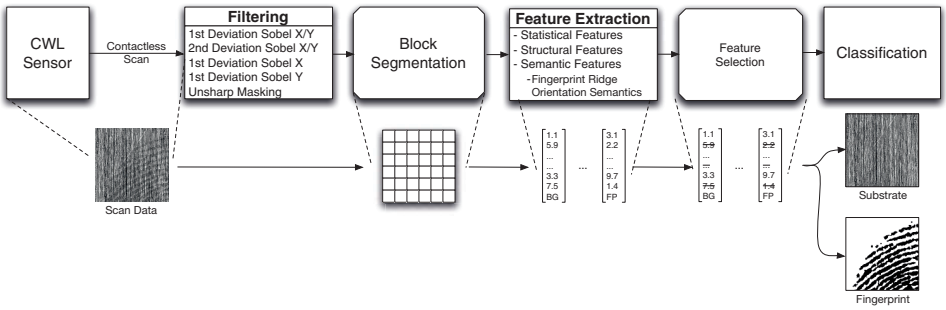


Figure 2: Data acquisition process, processing, and classification

3.1 Application scenario

The application scenario for this case study is the contact-less, non-invasive acquisition of latent fingerprints. The primary challenge of this technique is the inevitable acquisition of the substrate characteristics superimposing the fingerprint pattern. Depending on the substrate, the fingerprint can be rendered invisible. In order to allow for a forensic analysis of the fingerprint, it is necessary to differentiate between areas of the surface without fingerprint residue and others covered with fingerprint residue (fingerprint segmentation).

For this first evaluation, we solely rely on white furniture surface, because it provides a rather large difference between the substrate and the fingerprint. The achieved classification accuracy in a two-fold cross-validation based on 10 fingerprint samples is 93.1% for the J48 decision tree in [HKDV14]. The number of 10 fingerprints is sufficient for our evaluation, because we do not perform a biometric analysis. Due to the block-based classification, 1,003,000 feature vectors are extracted. For our extended 600 dimensional feature space (see Section 3.3), we achieve a classification accuracy of 90.008% based on 501,500 data sets for each of the two classes "*fingerprint*" and "*substrate*".

3.2 Data Gathering Process

The data gathering process utilizes a FRT CWL600 [Fri14] sensor mounted to a FRT MicroProf200 surface measurement device. This particular sensor exploits the effect of chromatic aberration of lenses to measure the distance and the intensity of the reflected light simultaneously. Due to this effect, the focal length of different wavelength is different. Thus, only one wavelength from the source of white light is focused at a time. This particular wavelength yields the highest intensity in the reflected light. So, it can be easily detected using a spectrometer by locating the maximum within the spectrum.

The intensity value is derived from the amplitude of this peak within the value range $[1; 4,095]$. The wavelength of the peak can be translated into a distance between the sensor and the measured object using a calibration table. The achieved resolution for this distance is 20 nm. The data itself are stored within a 16 bit integer array which can be afterward converted to a floating point distance value. The CWL600 is a point sensor which acquires the sample point-by-point while the sample is moved underneath. Thus, it is possible to select arbitrary lateral resolutions for the acquisition of the sample.

In our case study, we use a lateral dot distance of 10 μm which results in a resolution five times as high as the commonly used resolution of 500 ppi in biometric systems.

3.3 Data Description

The feature space in [HKDV14] contains statistical, structural, and fingerprint semantic features. The final feature space is extracted from the intensity and topography data (see

Section 3.2) and preprocessed versions of these data sets. Table 1 summarizes the 50 features which are extracted from each data set.

Feature Set	Features
Statistical Features	Minimum value; maximum value; span; mean value; median value; variance; skewness; kurtosis; mean squared error; entropy; globally and locally normalized values of absolute min, max, median; globally and locally normalized values of relative min, max, span, median; globally normalized absolute and relative mean value of B
Structural Features	Covariance of upper and lower half of a block B; covariance of left and right half of the block B; line variance of a block B; column variance of a block B; most significant digit frequency derived from Benford's Law [Ben38] (9 features); Hu moments [Hu62] (7 features)
Fingerprint Semantic Features	Maximum standard deviation in B_ϵ after Gabor filtering; mean value of the block B for the highest Gabor response

Table 1: Overview of the extracted features

All features are extracted from blocks with a size of 5×5 pixels with the exception of the fingerprint semantic feature of the maximum standard deviation in B_ϵ after Gabor filtering. The fingerprint semantic features are motivated by the fingerprint enhancement, e.g. [HWJ98], which utilize Gabor filters for emphasizing the fingerprint pattern after determining the local ridge orientation and frequency. Since this filtering relies on a ridge valley pattern, it requires larger blocks. In particular, we use a block size of 1.55 by 1.55 mm (155×155 pixels) as suggested in [HKDV14].

The features are extracted from the original and pre-processed data. In particular, the intensity and topography data are pre-processed using Sobel operators in first and second order in X and Y direction combined, Sobel operators in first order in X, as well as Y direction separately, and unsharp masking (subtraction of a blurred version of the data).

In result, we get a 600-dimensional feature space. However, some of the features cannot be determined, e.g., due to a division by zero in case of the relative statistical features. Thus, either the classifier must be able to deal with missing values, or those features need to be excluded. To this end, we apply the J48 classifier, because it handles missing data.

4 Evaluation

In this section, we present the evaluation of the classification according to the J48 algorithm. We restrict this study to performance measurements of computation time for

	Condition positive	Condition negative
Test outcome positive	True Positive (TP)	False Positive (FP)
Test outcome negative	False Negative (FN)	True Negatives (TN)

Table 2: Contingency table

building the model and for the evaluation of the model. As influences on the performance, we identify according to Section 2.2 cardinality of the dimensions and involved features. Therefore, we investigate the model quality with respect to precision and recall. First, we present our evaluation setup. This is followed by the result presentation. Finally, we discuss our findings.

4.1 Setup

Our data are preprocessed as described in Section 3. We use the implementation of C4.5 [Qui93] in WEKA, which is called J48. For the identification of relationships between included feature dimensions and feature cardinality and model build time and model evaluation time, we use different performance measurements regarding the model. We briefly describe the model performance measurements in the following.

In classification, the candidates can be classified correctly or incorrectly. Compared to the test population four cases are possible, as presented in Table 2.

In the following we define measures that can be derived from the contingency table. The recall (also called sensitivity or true positive rate) represents the correctly identified positive elements compared to all identified positive elements. This measure is defined as:

$$\mathit{Recall} = \frac{TP}{TP + FN} \tag{1}$$

Accuracy describes all correctly classified positive and negative elements compared to all elements. This measure assumes a non-skewed distribution of classes within the learning as well as training data. It is defined as:

$$\mathit{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{2}$$

Precision is also called positive prediction rate and measures all correctly identified positives compared to all positives in the ground truth. It is defined as:

$$\mathit{Precision} = \frac{TP}{TP + FP} \tag{3}$$

Specificity is also called true negative rate and is a ratio comparing the correctly classified negative elements to all negative classified elements. It is defined as:

$$\mathit{Specificity} = \frac{TN}{FP + TN} \tag{4}$$

The measure Balanced Accuracy is applied in the case that the classes are not equally distributed. This takes non-symmetric distributions into account. The balance is achieved by computing the arithmetic mean of Recall and Specificity and it is defined as:

$$\textit{Balanced Accuracy} = \frac{\textit{Recall} + \textit{Specificity}}{2} = \frac{1}{2} \cdot \left(\frac{TP}{TP + FN} + \frac{FN}{FP + TN} \right) \quad (5)$$

The F-Measure is the harmonic mean of precision and recall to deal with both interacting indicators at the same time. This results in:

$$\textit{F-Measure} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (6)$$

Depending on the application scenario, a performance measure can be used for optimization. In Fig. 3, we depict all above stated performance measurements according to a filtering of our data set.

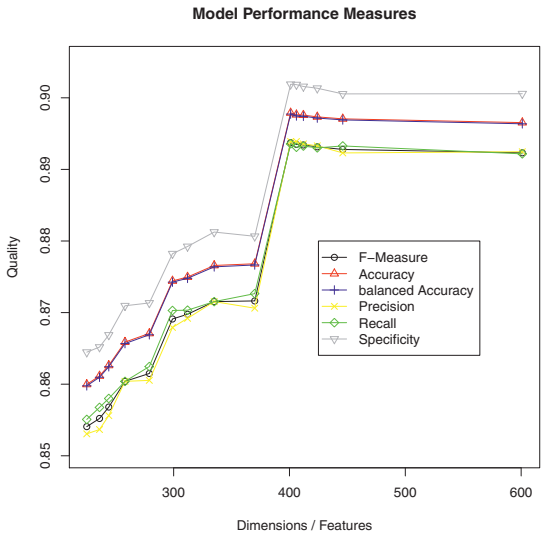


Figure 3: Performance Measures for different filters of the test case

In our evaluation, we investigate two different performance influences. On the one side, we are interested in filtering out correlated data columns. At the other side, we measure performance for a restricted data space domain. This is applied by a data discretization.

Evaluation is based on three important aspects:

- Building the model in terms of computation time,
- Testing the model in terms of computation time, and
- Quality of the model measured in model performance indicators.

From Fig. 3 it can be identified, that the computed models have a higher specificity than recall. This results also in a lower F-Measure. Furthermore, it can be seen that the training data are not imbalanced and accuracy is very close to balanced accuracy. However, all values are close and at an acceptable range. Therefore, we use for the remainder of our result presentation the F-Measure as model performance measure.

For reducing the dimensional space, we secondly discretize each feature. This is computed in such a way that the variance within a feature is retained as best as possible. Currently, there are no data structures within the WEKA environment, that use restricted data spaces efficiently. Therefore, we assume that model creation and model evaluation times are not significantly influenced. However, as a database system can be used in future, the question arises, which quality influence on model performance is achieved by discretization. Therefore, we conduct an evaluation series with discretized feature dimensions, where all feature dimensions are restricted to the following cardinalities:

- 8 values,
- 16 values,
- 32 values,
- 64 values,
- 128 values,
- 256 values,
- 512 values,
- 1,024 values,
- 2,048 values, and
- full cardinality.

4.2 Result Presentation

As a first investigation of our evaluation scenario, we present results regarding the elimination of features. For the feature elimination we decide for an statistical approach, where correlated data columns are eliminated from the data set.

In Fig. 4, we present the dimensions that are included in the data set. At the x-axis, we present the correlation criteria that are used for elimination. For instance, a correlation criteria of 99% means that all data columns are eliminated from the data set that have a correlation of 0.99 to another feature within the data set. Note, we compare every feature column with every other and at an elimination decision; we left the first in the data set. Therefore, we prefer the first data columns within the data set. Furthermore, we also tested the feature reduction for discretized data sets. With a small cardinality, the feature reduction due to correlation is lower, which means that the dimensional space is higher compared to the others.

We evaluate in the following the reduction of the feature space in terms of computational effort. We differentiate at this point two cases for this effort: On the one side the model building time represents computational performance for creating (learning) the model. As the amount of data tuples for learning the model we use 501,500 elements. As a second measurement, we present evaluation times where 501,500 further elements are used in a testing phase of the model. This additionally leads to the quality indicators of the model presented in Section 4.1. We present this information afterward.
In Fig. 5, we present the model creation times for different data sets. With a decrease of the feature space, the computation time reduces, too. However, there are some saltus identifi-

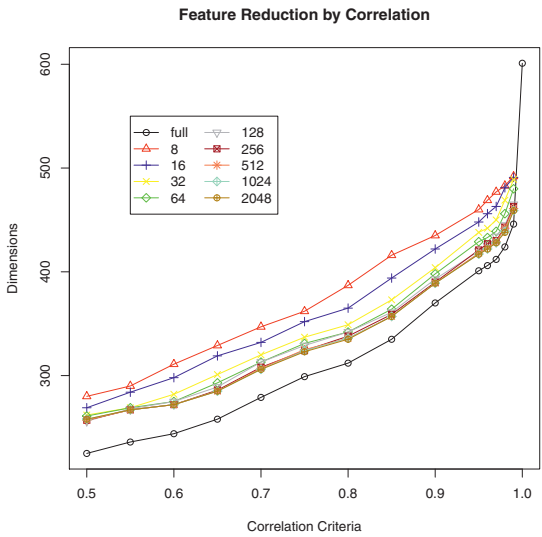


Figure 4: Feature Reduction by Correlation

able. These are related to the fact, that the algorithm has a dynamic model complexity. This means that the number of nodes within the model is not restricted and therefore, smaller models can have a faster generation time. Nevertheless, we do not focus on optimization for our case study, but we derive a general relationship. From our data, we can derive that a decrease is reduced for data that are not more than 85% correlated. This leads to a slower reduction in computation time. However, with this elimination of 85% correlated values, the computational effort is reduced to approximately one third. An important result from the model generation: a restrictive discretization (cases 8 and 16) does negatively influences the model building time. Note, we do not use in our evaluation an optimized data structure, which has a significant influence on the computational performance, see also Section 2.2. Although the underlying data structure is general, a restriction of the feature cardinalities improves model building times for the cases cardinality 32 and higher.

For evaluation times of the model a similar behavior is identifiable. In Fig. 6, we present the evaluation times for the same data sets. Two major differences can be easily seen: On the one side, the difference between the test cases is smaller and the slopes are smoother. On the other side, a reduction of the evaluation time is optimal for cardinalities of 32 and 64. An increase of the cardinalities leads to a higher computational effort. This is respected to the fact that the sequential searches within the data are quite important for the testing phase of a model. A usage of efficient data structures should therefore be in focus of future studies.

With both above presented evaluations, we only have computation time in the focus. However, we have to respect the quality of the model at the same time. Within classification applications, an increased information usage (in terms of data attributes) can increase the model quality. A reduction of the information space might lead to a lower model quality.

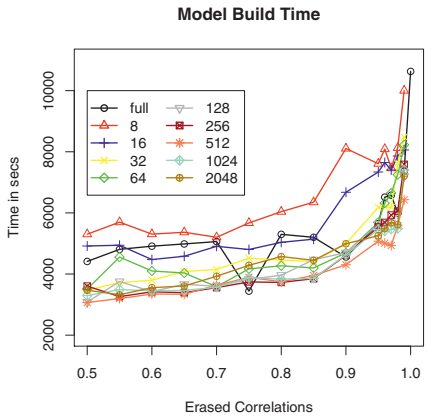


Figure 5: Model Build Time

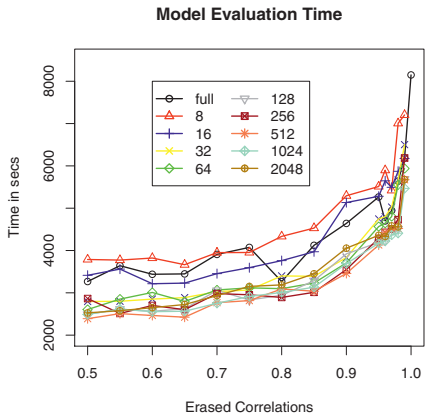


Figure 6: Model Evaluation Times

In Fig. 7, we show the relationship between erased correlations and the F-Measure. Note, that an increase in the F-Measure is also for a reduced data space possible (e.g., in the case of full cardinality). With a reduced cardinality in the information space, a lower F-Measure is achieved. This is especially true for low cardinalities (e.g., 8 or 16). However, in the case that the cardinality is reduced from a correlation of 0.95 to 0.9 within the data set a higher decrease in the F-Measure is identifiable. A second significant reduction of the F-measure is at the 0.7 correlation elimination level.

In Fig. 8, we present the relationship between model build times and the model quality. Although a negative dependency is assumed, this trend is only applicable to some parts of the evaluation space. As an optimization of model quality and computation time, the first high decrease model quality is at an elimination of 0.95 correlated values. Further eliminations do not influence the model build times in a similar decrease.

Overall, we have to state that our reduction of the data space is quite high compared to the reduction of the model quality in terms of the F-measure. Note, other model performance measures are quite similar.

4.3 Discussion

With our evaluation, we focus on the influences of the data space to model performance in terms of quality and computation times. Therefore, we reduce the information space in two ways. On the one hand, we restrict dimensionality by applying a feature reduction by correlation. This is also called canonical correlation analysis. It can be computed in a very efficient way and therefore, it is much faster than other feature reduction techniques, e.g., principal component analysis or linear discriminant analysis. Furthermore, we restrict the cardinality of the feature spaces, too. We discretize the feature space and are interested in

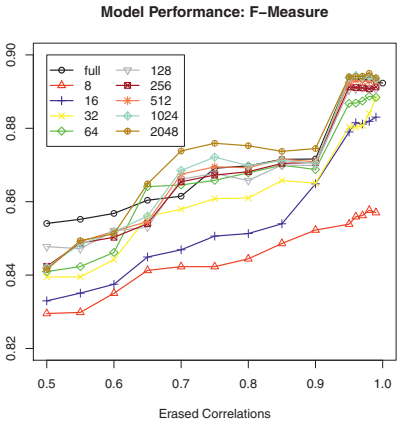


Figure 7: Model Quality and Reduction

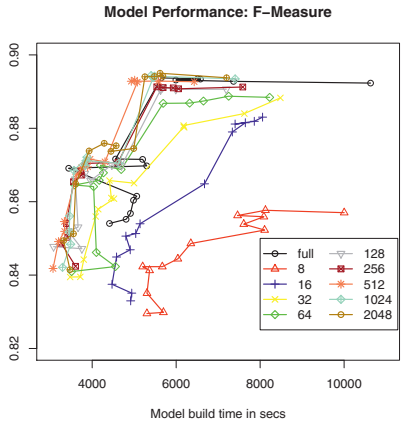


Figure 8: Model Quality and Build Times

the influence on model quality. An influence for the model build times are not assumed, due to the fact that the underlying data structures are not optimized.

We focus this in future work, cf. [BDKM14]. Due to the column-wise data processing of the classifiers, we assume that a change in the underlying storage structure, e.g., column stores or Data Dwarfs, leads to a significant computational performance increase. First analyzes of the WEKA implementation reveal a high integration effort. However, the benefits are very promising.

5 Conclusion

We present some ideas on improving model quality and computational performance for a classification problem. This work is a starting point to enhance the process with respect to optimize computation times in a biometric scenario. Additional use cases, e.g., indicator simulation [KL05], other data mining techniques [HK00], or operations in a privacy secure environment [DKK+14], can be applied to our main idea and have to be considered for filtering and reduction techniques. With our evaluation study we show that performance with respect to computation times as well as model quality can be optimized. However, a trade-off between both targets has to be achieved due to inter-dependencies.

In future work, we want to improve the process by integrating and optimizing the different steps. We assume, an efficient data access structure is beneficial for model computation times and therefore, increases the application scenario. However, this computational improvement relies on the information space, especially on dimensional cardinality and number of involved dimensions. With an easy to apply algorithm, a data processing enables a fast transformation of the feature space and smooth the way for more efficient data mining for forensic scenarios.

Acknowledgment

The work in this paper has been funded in part by the German Federal Ministry of Education and Research (BMBF) through the Research Program ”DigiDak+ Sicherheits-Forschungskolleg Digitale Formspuren” under Contract No. FKZ: 13N10818.

References

[AKZ08] Elke Achtert, Hans-Peter Kriegel, and Arthur Zimek. ELKI: A Software System for Evaluation of Subspace Clustering Algorithms. In *SSDBM*, LNCS (5069), pages 580–585. Springer, 2008.

[AMH08] Daniel J. Abadi, Samuel Madden, and Nabil Hachem. Column-Stores vs. Row-Stores: How different are they really? In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 967–980, Vancouver, BC, Kanada, 2008.

[BBK98] Stefan Berchtold, Christian Böhm, and Hans-Peter Kriegel. The Pyramid-technique: Towards Breaking the Curse of Dimensionality. *SIGMOD Rec.*, 27(2):142–153, 1998.

[BBK01] Christian Böhm, Stefan Berchtold, and Daniel A. Keim. Searching in High-dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases. *ACM Comput. Surv.*, 33(3):322–373, 2001.

[BDKM14] David Broneske, Sebastian Dorok, Veit Köppen, and Andreas Meister. Software Design Approaches for Mastering Variability in Database Systems. In *GvDB*, 2014.

[Ben38] Frank Benford. The Law of Anomalous Numbers. *Proceedings of the American Philosophical Society*, 78(4):551–572, 1938.

[Ben75] Jon Louis Bentley. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM*, 18(9):509–517, 1975.

[DKK+14] Jana Dittmann, Veit Köppen, Christian Krätzer, Martin Leuckert, Gunter Saake, and Claus Vielhauer. Performance Impacts in Database Privacy-Preserving Biometric Authentication. In Rainer Falk and Carlos Becker Westphall, editors, *SECURWARE 2014: The Eighth International Conference on Emerging Security Information, Systems and Technologies*, pages 111–117. IARA, 2014.

[Fri14] Fries Research & Technology GmbH. Chromatic White Light Sensor CWL, 2014. http://www.frt-gmbh.com/en/chromatic-white-light-sensor-frt-cwl.aspx.

[FSGM+98] Min Fang, Narayanan Shivakumar, Hector Garcia-Molina, Rajeev Motwani, and Jeffrey D. Ullman. Computing Iceberg Queries Efficiently. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB ’98, pages 299–310, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[GBS+12] Alexander Grebhahn, David Broneske, Martin Schäler, Reimar Schröter, Veit Köppen, and Gunter Saake. Challenges in finding an appropriate multi-dimensional index structure with respect to specific use cases. In Ingo Schmitt, Sascha Saretz, and Marcel Zierenberg, editors, *Proceedings of the 24th GI-Workshop ”Grundlagen von Datenbanken 2012”*, pages 77–82. CEUR-WS, 2012. urn:nbn:de:0074-850-4.

[GG98] Volker Gaede and Oliver Günther. Multidimensional Access Methods. *ACM Comput. Surv.*, 30:170–231, 1998.

[Gut84] Antonin Guttman. R-trees: A Dynamic Index Structure for Spatial Searching. *SIGMOD Rec.*, 14(2):47–57, 1984.

[HFH+09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10 – 18, 2009.

[HK00] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[HKDV14] Mario Hildebrandt, Stefan Kiltz, Jana Dittmann, and Claus Vielhauer. An enhanced feature set for pattern recognition based contrast enhancement of contact-less captured latent fingerprints in digitized crime scene forensics. In Adnan M. Alattar, Nasir D. Memon, and Chad D. Heitzenrater, editors, *SPIE Proceedings: Media Watermarking, Security, and Forensics*, volume 9028, pages 08/01–08/15, 2014.

[Hu62] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

[HWJ98] Lin Hong, Yifei Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):777 –789, aug 1998.

[KL05] Veit Köppen and Hans-J. Lenz. Simulation of Non-linear Stochastic Equation Systems. In S.M. Ermakov, V.B. Melas, and A.N. Pepelyshev, editors, *Proceeding of the Fifth Workshop on Simulation*, pages 373–378, St. Petersburg, Russia, July 2005. NII Chemistry Saint Petersburg University Publishers.

[KSS14a] Veit Köppen, Gunter Saake, and Kai-Uwe Sattler. *Data Warehouse Technologien*. MITP, 2 edition, Mai 2014.

[KSS14b] Veit Köppen, Martin Schäler, and Reimar Schröter. Toward Variability Management to Tailor High Dimensional Index Implementations. In *RCIS*, pages 452–457. IEEE, 2014.

[MKH+13] Andrey Makrushin, Tobias Kiertscher, Mario Hildebrandt, Jana Dittmann, and Claus Vielhauer. Visibility enhancement and validation of segmented latent fingerprints in crime scene forensics. In *SPIE Proceedings: Media Watermarking, Security, and Forensics*, volume 8665, 2013.

[Qui86] John Ross Quinlan. Induction of Decision Trees. *Mach. Learn.*, 1(1):81–106, 1986.

[Qui93] John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[SDRK02] Yannis Sismanis, Antonios Deligiannakis, Nick Roussopoulos, and Yannis Kotidis. Dwarf: Shrinking the PetaCube. In *SIGMOD*, pages 464–475. ACM, 2002.

[SGS+13] Martin Schäler, Alexander Grebhahn, Reimar Schröter, Sandro Schulze, Veit Köppen, and Gunter Saake. QuEval: Beyond high-dimensional indexing à la carte. *PVLDB*, 6(14):1654–1665, 2013.

[WB97] Roger Weber and Stephen Blott. An Approximation-Based Data Structure for Similarity Search. Technical Report ESPRIT project, no. 9141, ETH Zürich, 1997.

Using Different Encryption Schemes for Secure Deletion While Supporting Queries

Maik Schott, Claus Vielhauer, Christian Krätzer

Department Informatics and Media
Brandenburg University of Applied Sciences
Magdeburger Str. 50
14770 Brandenburg an der Havel, Germany
schott@fh-brandenburg.de
claus.vielhauer@fh-brandenburg.de

Department of Computer Science
Otto-von-Guericke-University Magdeburg
Universitaetsplatz 2
39106 Magdeburg, Germany
kraetzer@cs.uni-magdeburg.de

Abstract: As more and more private and confidential data is stored in databases and in the wake of cloud computing services hosted by third parties, the privacy-aware and secure handling of such sensitive data is important. The security of such data needs not only be guaranteed during the actual life, but also at the point where they should be deleted. However, current common database management systems to not provide the means for secure deletion. As a consequence, in this paper we propose several means to tackle this challenge by means of encryption and how to handle the resulting shortcomings with regards to still allowing queries on encrypted data. We discuss a general approach on how to combine homomorphic encryption, order preserving encryption and partial encryption as means of de-personalization, as well as their use on client-side or server-side as system extensions.

1 Introduction and state of the art

With the increase of data in general stored in databases especially its outsourcing into cloud services, privacy-related informations are also becoming more and more prevalent. Therefore there is an increasing need of maintaining the privacy, confidentiality, and in general security of such data. Additionally privacy is required by several national laws, like the Family Educational Rights and Privacy Act and the Health Insurance Portability and Accountability Act of the United States, the Federal Data Protection Act (Bundesdatenschutzgesetz) of Germany, or the Data Protection Directive (Directive 95/46/EC) of the European Union. All these legal regulations require the timely and guaranteed – in the sense that it is impossible to reconstruct – removal of private information. Such removal is called forensic secure deletion.

Aside from the regular challenges of this issue, e.g. the behavior of magnetic media to partially retain the state of their previous magnetization – leaving traces of data later overwritten with other data – or the wear-levelling techniques of solid-state memory media [Gu96], RAM or swap memory copies, and remote backups, database systems have an additional complexity due to their nature to provide an efficient and fast access to data by the means of introducing several redundancies of data. A certain information is not only stored within its respective database table, but also in other locations, like indexes, logs, result caches, temporary relations or materialized views [SML07]. Deleted rows are often just flagged as deleted, without touching the actually stored data. Additionally, due to page-based storage mechanisms, changes to records on these pages, requiring a change to the layout of these pages will, not necessarily update only this very page but instead create a new copy in the unallocated parts of the file system updating with the new data, but still leaving the original page behind flagged as unallocated space. The same applies to any kind of deletion operations. Essentially, old data is being marked as deleted, however remains present and is not immediately or intentionally deleted. The later happens only occasionally, when this unallocated space is later overwritten by a new page.

An extensive study of this issue was done by Stahlberg et al. [SML07] who forensically investigated five different database storage engines - IBM DB2, InnoDB, MyISAM (both MySQL), PostgreSQL, and SQLite - with regards to traces left of deleted data within the table storage, transaction log, indexes. They found that even after applying 25,000 operations and vacuuming, a large amount of deleted records could still be found. Furthermore they investigated the cost of overwriting or encrypting (albeit using highly insecure algorithms) log entries for the InnoDB engine. Grebhahn et al. [GSKS13] especially focused on index structures and what kind and amount of traces of deleted records can be reconstructed from the structure of indexes. Albeit they didn't investigate a real database system but a mockup designed to thoroughly evaluate high-dimensional indexes, they achieved recovery rates from R trees of up to 60% in single cases.

As shown, although a forensic secure deletion is required in many cases, the actual realization of removing data once it has been ingested to a database is still a difficult or even unsolved challenge. Therefore the solution must be sought at an earlier point: the time the data first enters the database. As the concept of deletion of data basically means rendering this data unreadable/illegible it shares similarities to the encryption of data without having knowledge of the proper key, as introduced by [BL96] for backup systems. Encryption can therefore be seen as a "preventive deletion" scheme in a forensic secure way.

At the same time the illegibility of encrypted data also hinders the widespread use in database systems as most operations and therefore queries on them are not possible compared to their unencrypted state. Therefore, in this paper we discuss a general approach on how to use several encryption schemes to provide additional security, while still maintaining some of the advantageous properties of unencrypted data. A similar concept has been introduced in the context of CryptDB e.g. in [PRZ+11] and more recently in [GHH+14].

2 Approach

Encryption schemes can generally be classified as symmetric and asymmetric, depending on if the encryption and decryption processes use the same shared key or different keys (public key/ and private key). As in asymmetric schemes the sender and receiver of a message use different keys, there is less trust required between both parties w.r.t. the key handling and secure storage; thus asymmetric schemes are more secure. However, this severely affects the performance and conversely symmetric schemes are several magnitudes faster than asymmetric algorithms.

In "traditional" (strong) cryptography, the goal is that the ciphertext does not reveal anything about the plaintext, e.g. any two given similar yet different plaintexts m_1 and m_2 the resulting ciphertexts are dissimilar and appear random. As such any operation which makes use of any property of the plaintext is not applicable to ciphertext. This is expressed by the cryptographic property is *ciphertext indistinguishability* introduced by [GM84] as *polynomial security*, i.e. given n plaintexts and their n ciphertexts, determining which ciphertexts refer to which plaintexts has the same probability as random guessing. Similar is the property non-malleability introduced by [DDN00] which states that given a ciphertext an attacker may not be able to modify this ciphertext in a way which would yield a related plaintext, i.e. a malleable scheme would fulfil: $E(m \oplus x) = E(m) \otimes x'$, with E being the encryption operation, m the plaintext, x a value to change the plaintext using the operation $\oplus$ and $x'/\otimes$ their counterparts in encrypted domain. However, there exist encryption schemes which intentionally give up on these security properties in exchange to provide for additional benefits, i.e. computational properties for mathematical operations in the encrypted domain.

The first one of these schemes is **Homomorphic Encryption** (HE), with its basic concept introduced by Rivest et al. [RAD78], as an encryption scheme allowing certain binary operations on the encrypted plaintexts in the ciphertext domain by a related homomorphic operation without any knowledge of the actual plaintext, i.e.: $E(m_1 \circ m_2) = E(m_1) \odot E(m_2)$. However, most of the early homomorphic encryption schemes only allowed one operation (multiplication or addition) until Gentry [Ge09] introduced the first *Fully Homomorphic Encryption* (FHE) scheme, which allow additive as well as multiplicative arithmetic operations at the same time. As can easily be seen, homomorphic encryption does not fulfil the non-malleability criterion, since every adversary can combine two ciphertexts to create a valid new encrypted plaintext. The main drawback of this encryption scheme is that, although Gentry's approach opened up lots of interest in the scientific community, the scheme it is very intensive with regards to computational time and space requirements, with a plaintext to ciphertext expansion factor of thousands to millions [LH14], and the computation of complex/chained operations may take several seconds.

The second encryption scheme is **Order-Preserving Encryption** (OPE) introduced by Agrawal et al. [AKSX04] and cryptographically proofed by Boldyreva et al. [BCLO09], who also provided a cryptographic model based on hypergeometric distribution. The idea of order-preservation encryption is to provide an encryption scheme which

maintains the property of order of plaintext within their encrypted counterparts, i.e. $\forall m_1 \geq m_2 : E(m_1) \geq E(m_2)$. It can be seen that this scheme does not fulfil the ciphertext indistinguishability criterion as by ordering the plaintexts and ciphertexts there is high probability of knowing which ciphertext can be mapped to which plaintext, except for equal plaintexts.

A third encryption scheme is **Partial Encryption,** differing from the aforementioned as it is not a specific new approach of encrypting, but a different application of existing common encryption schemes. The basic idea, as already discussed, e.g., in [MKD+11] and [SSM+11], is that in many complex data items only some parts are confidential. However, encrypting the complete data item may hinder the use of the data as even persons or processes who only want to access these non-confidential parts would need to get the data item decrypted in some way, e.g. by providing them with a higher security clearance, even though they should not need it. Using partial encryption, only the confidential parts of the data item are encrypted and associated with metadata specifying which parts. As the data is complex, and thus HE and OPE schemes would not be usable, partial encryption would use strong cryptography.

Combining these, our concept consists of two steps: The *first step* is to classify each data type with regards to its security level in the sense of maximum possible harm if the data gets misused or disclosed. This segmentation basically defines the amount of data to be protected as well as the type of security means needed to protect this data. Since the actual evaluation depends on the regulations of each organization, legislation, use-cases and so on, this is not the focus of this paper and thus will not be described in detail. The *second step* is to evaluate what kind of operations are commonly done to the data, in the sense of if the queries mostly consist of arithmetic operations, comparison operations, or other kinds of operations. If data is mainly used for arithmetic reasons, HE can be used to enable these kinds of operations on encrypted data. The same applies to data mainly used for comparison queries and OPE. If complex data types are present where only parts are sensitive and other parts contain useful information too, partial encryption can be used. It has to be stated here that this two-step concept is an extension of the basic approach discussed in [MKD+11].However, as stated before, the HE and OPE schemes are weaker than strong encryption schemes. Therefore, based on the evaluated security level of step 1, highly confidential data which satisfy the criteria for either HE or OPE should still be encrypted using strong cryptography.

3 Realization in database systems

In this section we describe the implications of using our approach on the database management system by discussing if the encryption or parts thereof should be provided server-side or client-side, as well as necessary changes to queries. For two of the introduced schemes (HE and OPE) the integration into a query language is discussed only conceptually on query level, while for the third scheme (PE) a realization of the query language extension required is presented.

Our test MySQL database consists of actual forensic data acquired during the Digi-Dak[1] project consisting of fiber scans, synthetic fingerprints and metadata in the form of statistical, spectral and gradient features for fingerprints [MHF+12] in 3.4 million tuples and diameter, length, perimeter, area, height and color as fiber features [AKV12].

3.1 Server-side vs client-side

An important notion is if the encryption E and decryption D functions should be provided client-side or server-side. On **client-side**, each client software would be required to implement both functions if it wants to create proper queries and interpret the results. However, this may be unfeasible for heterogeneous environments with many different types of client software, especially if future updates to the employed encryption are considered. The advantage of this approach is that the keys never leave the client.

If the crypto functions are centrally **server-based**, every client can make use of the encrypted data with only minor changes to the queries as described in the following sections. An important issue here is the acquisition of the keys. The keys may either be provided by the client or by the server. For *client-based key provision*, they would be part of the query and thus transmitted over the server-client connection. In this case this connection must be secured, e.g. by using SSL/TLS. For *server-based key provision* they may either be part of encryption or decryption functions themselves or are provided as part of a view as described in Section 3.4. The disadvantage *server-based key provision using views* approach is that the password needs to be stated in the view definition, and as such this encryption approach has at most the security level of the database access controls. As such it can protect confidential data against malicious clients or clients vulnerable to SQL injections and similar attacks, but not against attackers who have low-level (OS) access to the database. Therefore, client-based key provision has a higher security level, but as stated in Section 2 server-based key provision may also be feasible.

Approaches like [GSK13] propose an extension of the SQL syntax to permit special forensic tables that automatically handle secure deletion, and [SML07] proposes the use of an internal stream cipher to automatically encrypt every data. Both approaches need the extension/update of database systems on source code level which is in most cases not applicable to actual database systems in a productive environment. Therefore our approach consists of realizing secure deletion/encryption by making use of means provided by the database system, in our case external *User-Defined Functions* (UDF) – also called *call specification* in Oracle or *CLR Function* in MSSQL – from external libraries as complex cryptographic implementations are infeasible with stored procedures in SQL/PSM or related languages.

3.2 Order Preserving Encryption

Like for the other encryption schemes the data needs to be encrypted before it is stored in the database. This would be done by the client who transforms an input item m to

[1] http://omen.cs.uni-magdeburg.de/digi-dak/

$m' = E_{OPE}(m, key_E)$ which is then INSERTed. On a SELECT, the returned m' needs to be transformed back as $m = D_{OPE}(m', key_D)$. As described in the previous section, $E_{OPE}()$ and $D_{OPE}()$ may either be server- or client-side provided and the keys may either be explicitly provided or implicitly.

Assuming the columns of a tuple where c_2 contains confidential values mainly used for relational queries (for example with a fixed value v_1) a query in the form:

Query 1a: SELECT c_1, c_2 FROM t_1 WHERE $c_2 < v_1$;

would be transformed into the following for OPE data:

Query 1b: SELECT c_1, $D_{OPE}(c_2, key_D)$ FROM t_1 WHERE $c_2 < E_{OPE}(v_1, key_E)$;

In case the query result does not contain OPE data, the only overhead to non-encrypted data would be the encryption of the WHERE statement. Therefore the time complexity depends on the number and complexity of OPE statements in the SELECT statement multiplied by the result count and the number and complexity of OPE statements in the WHERE statement.

3.3 Homomorphic Encryption

For homomorphic encryption basically the same applies. Assuming the columns of a tuple where c_1 and c_2 contain confidential values mainly used for arithmetic operations queries to retrieve a value for a client or generate a value within the database:

Query 2a: SELECT $c_1 + c_2$ FROM t_1 WHERE ...;

Query 3a: INSERT INTO t_2 (c_3) SELECT $c_1 * c_2$ FROM t_1 WHERE ...;

would be transformed into the following for HE data:

Query 2b: SELECT $D_{HE}(ADD_{HE}(c_1, c_2), key_D)$ FROM t_1 WHERE ...;

Query 3b: INSERT INTO t_2 (c_3) SELECT $MUL_{HE}(c_1, c_2)$ FROM t_1 WHERE ...;

However depending on the homomorphic operations and if the query result is stored in the database or returned to the client the call to an additional function may not be needed. For example in the Paillier cryptosystem [Pa99] the addition of two plaintexts is expressed by a multiplication of the ciphertexts: $m_1 + m_2 \bmod n = m_1' * m_2' \bmod n^2$.

3.4 Partial Encryption

In our implementation the actual library and user-defined functions were written in C#, as its runtime library provides a large amount of conveniently usable image processing and cryptography functions. As C# libraries use different export signatures the Unmanaged Exports tool (MIT license) by Robert Giesecke[2] to automatically create proper C style exports and unmarshalling.

The pseudo-code of the decrypt UDF called D_{PE} is as follows:

```
INPUT: blob, password
OUTPUT: image
```

[2] https://sites.google.com/site/robertgiesecke/Home/uploads/unmanagedexports

```
(image, regions, cryptalg, cryptparams, blocksize) ← unpack(blob)
if password = ∅ then return image endif
buffer ← ARRAY BYTE[1..blocksize], coord ← ARRAY POINT[1..blocksize]
b ← 0, h ← height(image), w ← width(image)
for y = 1 to h do
for x = 1 to w do
   if (x, y) in regions then
      buffer[b] ← image[x, y], coord[b] ← (x, y)
      b ← b+1
   endif
   if b = blocksize OR (y=h AND x=w)) then
      buffer' ← decrypt(buffer, cryptalg, cryptparams, password)
      for i = 1 to blocksize do image[coord[i]] = buffer'[i] endfor
      b ← 0
   endif endfor endfor
return image
```

In our scenario partial encryption is used for fingerprint scans. At a crime scene there are cases where latent fingerprints – sensitive data – may be superimposed with other non-sensitive evidence like fiber traces. Depending on the investigation goal (fingerprint of fiber analysis) it may thus become necessary to make areas containing sensitive data inaccessible. Therefore in our approach, only the fingerprint parts of the scans are encrypted by using AES, packed in ZIP container along the encryption metadata (algorithm, bit size, initialization vectors, password salt) and the outline of partially encrypted regions as shown in Figure 1.

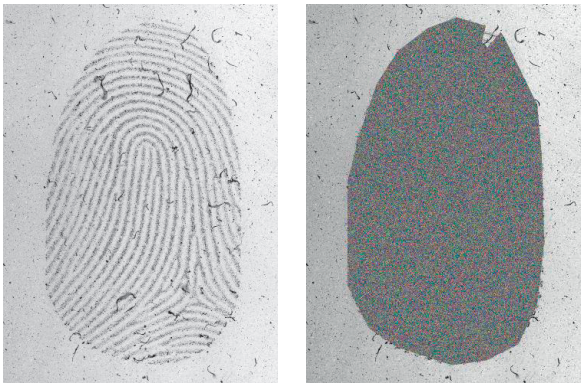


Figure 1: (Synthetic) original[3] and partially encrypted fingerprint

[3] Example from the Public Printed Fingerprint Data Set – Chromatic White Light Sensor - Basic Set V1.0. The image was acquired using sensory from the Digi-Dak research project (http://http://omen.cs.uni-magdeburg.de/digi-dak/, 2013) sponsored by the German Federal Ministry of Education and Research, see publication: Hildebrandt, M, Sturm, J., Dittmann, J., and Vielhauer, C.: Creation of a Public Corpus of Contact-Less Acquired Latent Fingerprints without Privacy Implications. Proc. CMS 2013, Springer, LNCS 8099, 2013, pp. 204–206; it uses privacy implication free fingerprint patterns generated with SFinGe, published in Cappelli, R.: Synthetic fingerprint generation, Maltoni, D., Maio, D., Jain, A.K., and Prabhakar, S. (Eds.): Handbook of Fingerprint Recognition, 2nd edn., Springer London, 2009.

The database itself only stores the container. Additionally for *server-based key provision using views* there would be two views to provide standardized means of access on the database query language level: a) an "anonymized view" for general users who do not have the proper security clearance to access unencrypted fingerprints, returning the encrypted fingerprint scan from the container (Figure 1 right side):

Query 4: CREATE VIEW fingerprints_anon AS SELECT id, filename, D_{PE}(scan) AS scan FROM fingerprints;

And b) a "deanonymized view" for privileged users like forensic fingerprint experts who do have the access rights to the actual fingerprints not only unpack the image but also try to decrypt it with the provided key (Figure 1 left side):

Query 5: CREATE VIEW fingerprints_deanon AS SELECT id, filename, D_{PE}(scan, *key*) AS scan FROM fingerprints;

Regarding the performance on our test system (Intel Core i7-4610QM @ 2.3GHz, 8 GB RAM), we observe in first experiments that querying 1000 tuples of containers takes 0.607s, unpacking the encrypted scans (Query 4) 29.110s, and returning the deanonymized scans (Query 5) 99.016s. However, it should be noted that for the Query 5 task, computation power for image parsing is included in the given figure and makes up for the main part of execution time for this query.

4 Conclusion and future work

In this paper we showed a general concept on how to use different encryption schemes for the sake of secure deletion and de-personalization, also taking into account that the data still at least partially remains usable for queries. The concept includes of the classification of data by its security level and later main usage which decides the appropriate encryption scheme. We also showed how these encryption schemes can be used in common database systems without the need to directly modify the system, but using its existing capabilities of a selected database system.

In future work a more thorough research on key management for *server-based key provision* needs to be done, as well the applicability to other database systems. Furthermore, the actual performance impact of this general concept to practical systems has to be evaluated with large-scale evaluations for relevant application scenarios, like e.g. larger forensic databases and/or biometric authentication systems where such a scheme could prevent information leakage as well as inter-system traceability.

5 Acknowledgements

The work in this paper has been funded by the Digi-Dak project by the German Federal Ministry of Education and Science (BMBF) through the research program under the contract no. FKZ: 13N10816.

References

[AKSX04] Agrawal, R.; Kiernan, J.; Srikant, R.; Xu, Y.: Order-preserving encryption for numeric data. In: SIGMOD, 2004; pages 563–574.

[AKV12] Arndt, C.; Kraetzer C.; Vielhauer, C.: First approach for a computer-aided textile fiber type determination based on template matching using a 3D laser scanning microscope. In: Proc. 14th ACM Workshop on Multimedia and Security, 2012.

[BCLO09] Boldyreva, A., Chenette, N., Lee, Y., O'Neill, A.: Order-preserving symmetric encryption. In: EUROCRYPT, 2009, pages 224–241.

[BL96] Boneh, D.; Lipton, R. J.: A revocable backup system. In: USENIX Security Symposium, 1996; pages 91–96.

[DDN00] Dolev, D.; Dwork, C.; Naor, M.: Nonmalleable Cryptography. In: SIAM Journal on Computing 30 (2), 2000; pages 391–437.

[Ge09] Gentry, C: Fully homomorphic encryption using ideal lattices. In: ACM symposium on Theory of computing, STOC '09, New York, 2009; pages 169–178.

[GM84] Goldwasser, S.; Micali, S.: Probabilistic encryption. In: Journal of Computer and System Sciences. 28 (2), 1984; pages 270–299.

[GSK13] Grebhahn, A.; Schäler, M.; Köppen, V.: Secure Deletion: Towards Tailor-Made Privacy in Database Systems. In: Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS), BTW, Köllen-Verlag, 2013; pages 99–113.

[GSKS13] Grebhahn, A.; Schäler, M.; Köppen, V.; Saake, G.: Privacy-Aware Multidimensional Indexing. In: BTW 2013; pages 133–147.

[GHH+14] Grofig, P.; Hang, I.; Härterich, M.; Kerschbaum, F.; Kohler, M.; Schaad, A.; Schröpfer, A.; Tighzert, W.: Privacy by Encrypted Databases. Preneel, B.; Ikonomou, D. (eds.): Privacy Technologies and Policy. Springer International Publishing, Lecture Notes in Computer Science, 8450, ISBN: 978-3-319-06748-3, pp. 56-69, 2014.

[Gu96] Gutmann, P.: Secure Deletion of Data from Magnetic and Solid-State Memory. In: USENIX Security Symposium, 1996.

[MHF+12] Makrushin, A.; Hildebrandt, M.; Fischer, R.; Kiertscher, T.; Dittmann, J.; Vielhauer, C.: Advanced techniques for latent fingerprint detection and validation using a CWL device. In: Proc. SPIE 8436, 2012; pages 84360V.

[MKD+11] Merkel, R.; Kraetzer, C.; Dittmann, J.; Vielhauer, C.: Reversible watermarking with digital signature chaining for privacy protection of optical contactless captured biometric fingerprints - a capacity study for forensic approaches. Proc. 17th International Conference on Digital Signal Processing (DSP), 2011.

[LN14] Lepoint, T.; Nehring, A.: A Comparison of the Homomorphic Encryption Schemes FV and YASHE. Africacrypt 2014.

[Pa99] Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Eurocrypt 99, Springer Verlag, 1999; pages 223–238.

[PRZ+11] Popa, R. A.; Redfield, C. M. S.; Zeldovich, N.; Balakrishnan, H.: CryptDB: Protecting Confidentiality with Encrypted Query Processing. Proc. 23rd ACM Symposium on Operating Systems Principles (SOSP), 2011.

[RAD78] Rivest, R. L.; Adleman, L.; Dertouzos, M. L.: On data banks and privacy homomorphisms. In: Foundations of Secure Computation, 1978.

[SSM+11] Schäler, M.; Schulze, S.; Merkel, R.; Saake, G.; Dittmann, J.: Reliable Provenance Information for Multimedia Data Using Invertible Fragile Watermarks. In Fernandes, A. A. A.; Gray, A. J. G.; Belhajjame, K. (eds.): Advances in Databases. Springer Berlin Heidelberg, Lecture Notes in Computer Science, 7051, pp. 3-17, ISBN: 978-3-642-24576-3, 2011.

[SML07] Stahlberg, P.; Miklau, G.; Levine, B.N.: Threats to Privacy in the Forensic Analysis of Database Systems. In: SIGMOD, New York 2007, ACM; pages 91–102.

Data Streams and Event Processing

Data Streams and Event Processing

Marco Grawunder[1], marco.grawunder@uni-oldenburg.de
Daniela Nicklas[2], daniela.nicklas@uni-bamberg.de

[1]Universität Oldenburg
[2]Universität Bamberg

The processing of continuous data sources has become an important paradigm of modern data processing and management, covering many applications and domains such as monitoring and controlling networks or complex production system as well complex event processing in medicine, finance or compliance.

- Data streams
- Event processing
- Case Studies and Real-Life Usage
- Foundations
  - Semantics of Stream Models and Languages
  - Maintenance and Life Cycle
  - Metadata
  - Optimization
- Applications and Models
  - Statistical and Probabilistic Approaches
  - Quality of Service
  - Stream Mining
  - Provenance
- Platforms for event and stream processing, in particular
  - CEP Engines
  - DSMS
  - ”Conventional” DBMS
  - Main memory databases
  - Sensor Networks
- Scalability
  - Hardware acceleration (GPU, FPGA, ...)
  - Cloud Computing
- Standardisation

In addition to regular workshop papers, we invite extended abstracts to cover hot topics, ongoing research and ideas that are ready to share and discuss, but maybe not ready to publish yet.

1 Workshop co-chairs

Marco Grawunder (Universität Oldenburg)
Daniela Nicklas (Universität Bamberg)

2 Program Committee

Andreas Behrend (Universität Bonn)
Klemens Boehm (Karlsruher Institut für Technologie)
Peter Fischer (Universität Freiburg)
Dieter Gawlick (Oracle)
Boris Koldehofe (Technische Universität Darmstadt)
Wolfgang Lehner (TU Dresden)
Richard Lenz (Universität Erlangen-Nürnberg)
Klaus Meyer-Wegener (Universität Erlangen)
Gero Mühl (Universität Rostock)
Kai-Uwe Sattler (Technische Universität Ilmenau)
Thorsten Schöler (Hochschule Augsburg)

Modulares Verteilungskonzept für Datenstrommanagementsysteme

Timo Michelsen, Michael Brand, H.-Jürgen Appelrath

Universität Oldenburg, Department für Informatik
Escherweg 2, 26129 Oldenburg
{timo.michelsen, michael.brand, appelrath}@uni-oldenburg.de

Abstract: Für die Verteilung kontinuierlicher Anfragen in verteilten Datenstrommanagementsytemen (DSMS) gibt es je nach Netzwerk-Architektur und Anwendungsfall unterschiedliche Strategien. Die Festlegung auf eine Strategie ist u. U. nachteilig, besonders wenn sich Netzwerk-Architektur oder Anwendungsfall ändern. In dieser Arbeit wird ein Ansatz für eine flexible und erweiterbare Anfrageverteilung in verteilten DSMSs vorgestellt. Der Ansatz umfasst drei Schritte: (1) Partitionierung, (2) Modifikation und (3) Allokation. Bei der Partitionierung wird eine kontinuierliche Anfrage in disjunkte Teilanfragen zerlegt. Die optionale Modifikation erlaubt es, Mechanismen wie Fragmentierung oder Replikation zu verwenden. Bei der Allokation werden die einzelnen Teilanfragen schließlich Knoten im Netzwerk zugewiesen, um dort ausgeführt zu werden.

Für jeden der drei Schritte können unabhängige Strategien verwendet werden. Dieser modulare Aufbau ermöglicht zum Einen eine individuelle Anfrageverteilung. Zum Anderen können bereits vorhandene Strategien aus anderen Arbeiten und Systemen (z.B. eine Allokationsstrategie) integriert werden. In dieser Arbeit werden für jeden der drei Teilschritte beispielhafte Strategien vorgestellt. Außerdem zeigen zwei Anwendungsbeispiele die Vorteile des vorgestellten, modularen Ansatzes gegenüber einer festen Verteilungsstrategie.

1 Einleitung

In verarbeitenden Systemen ist es häufig notwendig, den persistenten Teil des Systems mehrfach und verteilt vorzuhalten, um Ausfälle kompensieren zu können. In verteilten Datenbankmanagementsystemen (DBMS) werden persistente Daten disjunkt (Fragmentierung) oder redundant (Replikation) auf verschiedene Knoten des Netzwerks verteilt. Anfragen werden einmalig gestellt und greifen ausschließlich auf die Knoten zu, die die betreffenden Daten besitzen.

Betrachtet man allerdings verteilte Datenstrommanagementsysteme (DSMS), so ist eine Verteilung der Datenstromelemente allein nicht zielführend, da diese flüchtig sind. Hier sind die Anfragen persistent, da sie theoretisch kontinuierlich ausgeführt werden. Eine solche, auf mehrere Knoten verteilte, *kontinuierliche Anfrage* wird *verteilte, kontinuierliche Anfrage* genannt. Eine verteilte, kontinuierliche Anfrage befindet sich auf mehreren Knoten, indem die einzelnen Operationen den Knoten zugeordnet und dort ausgeführt

werden (in sogenannten *Teilanfragen*). Zwischenergebnisse der Teilanfragen werden von Knoten zu Knoten gesendet, bis schließlich die Ergebnisse an den Nutzer gesendet werden. Für eine Zerlegung einer kontinuierlichen Anfrage in Teilanfragen gibt es allerdings viele Möglichkeiten (z.B. eine Teilanfrage für die gesamte kontinuierliche Anfrage oder je eine Teilanfrage pro Operation).

Die Zerlegung in Teilanfragen ist nur ein Aspekt, der bereits aufzeigt, dass unterschiedliche Strategien für eine Anfrageverteilung in verteilten DSMSs existieren. Ein weiterer Aspekt ist die Netzwerk-Architektur (bspw. homogene Cluster, Client/Server-Architekturen und Peer-To-Peer (P2P)-Netzwerke). In einem homogenen Cluster verfügen alle Knoten i.d.R. über die gleichen Mengen an Systemressourcen. Daher muss bei einer Zuweisung von (Teil-) Anfragen an einen Knoten in homogenen Clustern keinerlei Heterogenität berücksichtigt werden. Kommt hingegen ein heterogenes P2P-Netzwerk zum Einsatz, kann es vorkommen, dass nicht jeder Knoten jede (Teil-) Anfrage ausführen kann (bspw. aufgrund komplexer Operatoren).

In vielen verteilten DSMSs ist die Anfrageverteilung system-intern auf eine bestimmte Netzwerk-Architektur ausgelegt (z.B. Client/Server). Somit wird es aufwendig, die zugrunde liegende Netzwerk-Architektur nachträglich zu wechseln. Ein Grund für eine solche Einschränkung ist die Tatsache, dass keine optimale Verteilungsstrategie existiert, die für alle Netzwerk-Architekturen und Knoten eingesetzt werden kann. Neben unterschiedlichen Netzwerk-Architekturen können ebenso anwendungsspezifische Kenntnisse des Nutzers die Anfrageverteilung optimieren (z.B. die Identifikation von Operatoren, die eine hohe Systemlast erzeugen). Eine solche manuelle Optimierung ist allerdings nur möglich, wenn der Nutzer die Anfrageverteilung für eine konkrete kontinuierliche Anfrage konfigurieren kann.

Diese Arbeit stellt ein Konzept für eine flexible und erweiterbare Anfrageverteilung in einem verteilten DSMS vor. Dazu wird die Verteilung in drei Phasen unterteilt. Jede Phase bietet eine eindeutige Schnittstelle mit Ein- und Ausgaben, die die Umsetzung mehrerer Strategien ermöglicht. Die Anfrageverteilung kann individuell für die zugrunde liegende Netzwerk-Architektur und den konkreten Anwendungsfall konfiguriert und ggfs. optimiert werden. Außerdem ist es möglich, bereits existierende Strategien aus anderen Quellen zu übernehmen. Es entsteht eine Sammlung an Strategien, die je nach Anwendungsfall individuell kombiniert werden können.

Der Rest der Arbeit ist wie folgt gegliedert: Abschnitt 2 gibt einen Überblick über andere Systeme, die ebenfalls kontinuierliche Anfragen verteilen. Das modulare Verteilungskonzept wird in Abschnitt 3 vorgestellt. Dabei liegt der Fokus auf der Erläuterung des Konzeptes. Auf eine vollständige Übersicht aller zum jetzigen Zeitpunkt verfügbarer Strategien wird in dieser Arbeit verzichtet. Abschnitt 4 beinhaltet den aktuellen Stand der Implementierung und beispielhafte Anwendungsszenarien. In Abschnitt 5 wird die Arbeit abschließend zusammengefasst.

2 Verwandte Arbeiten

In den DSMSs Borealis [CBB+03] und StreamGlobe [KSKR05] werden neue kontinuierliche Anfragen zunächst grob zerlegt und im Netzwerk verteilt. Häufig werden einzelne Operatoren Knoten zugeordnet. Nach der anfänglichen Verteilung werden Teile der kontinuierlichen Anfrage während der Verarbeitung verschoben, d.h., sie werden von Knoten zu Knoten übertragen. Dadurch können Kommunikationskosten gespart werden, indem bspw. Teilanfragen verschiedener Knoten zusammengefasst werden.

Das DSMS Stormy [LHKK12] zerlegt keine kontinuierlichen Anfragen, sondern führt sie stets vollständig auf einem Knoten aus. Die Auswahl des Knotens geschieht mittels eines Hashwertes, welcher aus der kontinuierlichen Anfrage gebildet wird: Jeder Knoten übernimmt einen Teil des Wertebereichs, sodass kontinuierliche Anfragen mit bestimmten Hashwerten bestimmten Knoten zugeordnet werden.

StreamCloud [GJPPM+12], Storm [TTS+14] und Stratosphere [WK09] nutzen Cloud-Infrastrukturen, um bei Bedarf verarbeitende, virtuelle Knoten in der Cloud zu erzeugen und Teilanfragen zuzuordnen. In Storm kann der Nutzer zusätzlich angeben, wie viele Instanzen für eine Operation erstellt werden sollen (bspw. für Fragmentierung oder Replikation). StreamCloud bietet unterschiedliche Zerlegungsstrategien, die später in dieser Arbeit aufgegriffen werden.

In SPADE, einer Anfragesprache von System S [GAW09], werden kontinuierliche Anfragen nach einem Greedy-Algorithmus zerlegt. Die Zuweisung der so entstandenen Teilanfragen wird mittels eines Clustering-Ansatzes durchgeführt. Dadurch sollen die Teilanfragen möglichst wenigen Knoten zugeordnet werden.

Daum et al. [DLBMW11] haben ein Verfahren für eine automatisierte, kostenoptimale Anfrageverteilung vorgestellt. Sie verfolgen damit andere Ziele als das in dieser Arbeit vorgestellte Konzept, bei dem es viel mehr um Flexibilität und Modularität als um Automatisierung geht.

Jedes hier vorgestellte DSMS besitzt seine eigene Vorgehensweise (mit eigenen Vor- und Nachteilen), kontinuierliche Anfragen im Netzwerk zu verteilen. Jedoch bieten sie – im Gegensatz zu dem hier vorgestellten Konzept – nicht die Flexibilität und Modularität, um Verteilungsstrategien bei Bedarf zu wechseln. Sie erlauben häufig nur mit großem Aufwand neue, evtl. domänenspezifische Strategien zu implementieren und einzusetzen.

3 Konzept

Häufig werden kontinuierliche Anfragen vom Nutzer deklarativ gestellt. Anschließend wird der Anfragetext i.d.R. in eine sprachenunabhängige Struktur übersetzt, die sich an Anfragepläne von DBMSs orientiert: die sogenannten *logischen Operatorgraphen*. Sie können als gerichtete, azyklische Graphen interpretiert werden, wobei die Knoten die Operatoren repräsentieren. Die Kanten stellen die Datenströme zwischen den Operatoren dar. Ein *logischer Operator* beschreibt, welche Operation auf einen Datenstrom ausgeführt

werden soll (bspw. Selektion, Projektion, Join). Sie beinhaltet jedoch nicht die konkrete Implementierung. Diese wird erst bei der tatsächlichen Ausführung der (Teil-) Anfragen auf einem Knoten eingesetzt. Aufgrund der Unabhängigkeit von der Sprache und von der Implementierung basiert das hier vorgestellte Konzept auf logischen Operatorgraphen. Es beschreibt also, wie die Verteilung eines logischen Operatorgraphen flexibel und erweiterbar durchgeführt werden kann.

Wie bereits erwähnt, ist eine einzige, fest programmierte Vorgehensweise zur Anfrageverteilung häufig nicht praktikabel. Dementsprechend wird in dieser Arbeit eine mehrschrittige Vorgehensweise verfolgt: (1) Partitionierung, (2) Modifikation und (3) Allokation. Abbildung 1 soll den Zusammenhang der Phasen zueinander verdeutlichen. Für jede Phase werden mehrere Strategien zur Verfügung gestellt, aus denen der Nutzer für jede kontinuierliche Anfrage wählen kann. Als Alternative ist eine automatisierte Auswahl vorstellbar, diese wird jedoch in dieser Arbeit nicht weiter verfolgt.

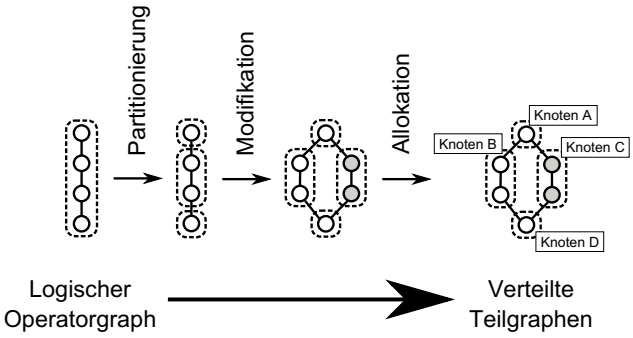


Abbildung 1: Phasen der Verteilung kontinuierlicher Anfragen.

Ausgehend von einem logischen Operatorgraphen wird in der ersten Phase, der *Partitionierung*, der Graph in *Teilgraphen* zerlegt, wobei keinerlei Änderungen am Graphen vorgenommen werden (z.B. neue Operatoren). Diese Teilgraphen werden dann *modifiziert*, um weitere Eigenschaften im Graphen sicherzustellen. Dabei können die Teilgraphen ergänzt, verändert oder auch entfernt werden. Beispielsweise wurde in der Abbildung der mittlere Teilgraph repliziert. Anschließend wird in der *Allokation* eine Zuordnung zwischen Teilgraph und ausführenden Knoten hergestellt und der Teilgraph schließlich übermittelt. In dieser Phase werden die Teilgraphen nicht mehr verändert.

```
1 #NODE_PARTITION <Partitionierungsstrategie> <Parameter>
2 #NODE_MODIFICATION <Modifikationsstrategie> <Parameter>
3 #NODE_ALLOCATE <Allokationsstrategie> <Parameter>
4
5 kontinuierliche Anfrage
```

Listing 1: Selektion der Strategien zur Verteilung einer kontinuierlichen Anfrage.

Die Auswahl der Strategien wird hier dem Nutzer überlassen. Ein Beispiel, wie der Nutzer eine kontinuierliche Anfrage im DSMS Odysseus [AGG+12] verteilen und entsprechende

Strategien auswählen kann, ist in Listing 1 zu sehen. Zu Beginn gibt der Nutzer die Strategie für jede Phase an (Zeilen 1 bis 3). Mittels Parameter können die Strategien weiter verfeinert werden (bspw. Replikationsgrad). Anschließend wird die eigentliche Anfrage formuliert, welche anhand der gewählten Kombination der Strategien verteilt wird.

Der Nutzer muss den Anfragetext nicht speziell anpassen, um diese verteilen zu können. Es müssen lediglich zuvor die eingesetzten Strategien spezifiziert werden. Für eine andere Anfrage können wiederum andere Strategien gewählt werden. In den folgenden Abschnitten werden die Phasen zur Verteilung genauer beschrieben und beispielhafte Strategien kurz erläutert.

3.1 Partitionierung

Aufgrund der Tatsache, dass eine gegebene kontinuierliche Anfrage in einem Netzwerk verteilt werden soll, werden einzelne Knoten häufig Teile der Anfrage erhalten. Dementsprechend muss zuvor entschieden werden, wie die Anfrage zerlegt werden soll. Das bedeutet in diesem Fall, dass der logische Operatorgraph in mehrere Teilgraphen zerlegt werden muss. Teilgraphen beschreiben somit, welche logischen Operatoren zusammen auf einem Knoten ausgeführt werden sollen. Es ist dabei wichtig, dass die Teilgraphen zueinander disjunkt sind, d.h., jeder logischer Operator der kontinuierlichen Anfrage befindet sich in genau einem Teilgraphen. Die logischen Operatoren müssen innerhalb eines Teilgraphen jedoch nicht zusammenhängend sein.

Das Finden einer geeigneten Zerlegung eines Graphen ist NP-hart [BMS+13]. Dementsprechend wird vorgeschlagen, mehrere Strategien zur Partitionierung anzubieten. Eine *Partitionierungsstrategie* erhält einen logischen Operatorgraphen und liefert eine Menge an disjunkten Teilgraphen. Einige Partitionierungsstrategien sind die folgenden:

QueryCloud Der logische Operatorgraph wird als ein Teilgraph behandelt. Das bedeutet, dass keine Zerlegung durchgeführt wird. Dies ist nützlich, wenn es sich um eine Anfrage mit einer geringen Zahl an logischen Operatoren handelt und eine Zerlegung nicht praktikabel erscheint.

OperatorCloud Jeder logischer Operator ist ein Teilgraph. Dies repräsentiert die maximale Zerlegung des Operatorgraphen. Diese Strategie ist für kontinuierliche Anfragen interessant, die wenige, jedoch sehr komplexe logische Operatoren beinhalten.

OperatorSetCloud In vielen kontinuierlichen Anfragen verursachen die zustandsbehafteten Operatoren die meiste Systemlast, wie bspw. Aggregationen [GJPPM+12]. Die Partitionierungsstrategie *OperatorSetCloud* zerlegt den logischen Operatorgraphen, sodass jeder Teilgraph maximal einen solchen Operator enthält. Dadurch ist es möglich, die zustandsbehafteten Operatoren verschiedenen Knoten zuzuteilen, sodass die Systemlast besser im Netzwerk verteilt werden kann.

Nutzerbasiert Falls es die Anfragesprache erlaubt, kann der Nutzer direkt angeben, welche Operatoren zusammen auf einem Knoten ausgeführt werden sollen. Diese Strategie ist besonders für Evaluationen praktisch, da damit bestimmte Szenarien der Verteilung nachgestellt und reproduziert werden können.

Auf Details wird im Rahmen dieser Arbeit verzichtet. Die ersten drei Strategien wurden dem Vorbild des DSMS StreamCloud nachempfunden und werden in [GJPPM+12] genauer erläutert. Es ist möglich, dass Entwickler weitere Strategien konzipieren und einsetzen. Dadurch können in konkreten Anwendungsszenarien bspw. spezielle Eigenschaften der Knoten und des Netzwerks ausgenutzt werden.

3.2 Modifikation

In der zweiten Phase wird der logische Operatorgraph modifiziert, um weitere Eigenschaften in der kontinuierlichen Anfrage sicherzustellen. Dieser Schritt ist optional und kann übersprungen werden, wenn keine Änderungen am logischen Operatorgraphen notwendig sind. Sind jedoch mehrere Modifikationen notwendig, kann diese Phase mehrfach durchgeführt werden. An dieser Stelle sind ebenfalls unterschiedliche Möglichkeiten vorstellbar. Jede *Modifikationsstrategien* erhält als Eingabe die Menge an Teilgraphen aus der ersten Phase. Die Ausgabe beinhaltet eine modifizierte Menge an Teilgraphen.

In dieser Phase ist ebenfalls vorgesehen, dass Entwickler weitere Strategien konzipieren und einsetzen. Jedoch wurden in der vorliegenden Arbeit folgende Modifikationsstrategien betrachtet:

Replikation Jeder Teilgraph wird (u. U. mehrfach) repliziert. Dadurch kann jeder Teilgraph auf mehreren Knoten ausgeführt werden. Solange mindestens ein Knoten den Teilgraphen ausführt, können (Zwischen-) Ergebnisse berechnet und gesendet werden.

Horizontale Fragmentierung Ähnlich zur Replikation wird jeder Teilgraph repliziert, jedoch empfängt jede Kopie nur einen Teil des Datenstroms. Die Ergebnisse werden am Ende wieder zusammengefasst. Damit kann die Verarbeitung parallelisiert werden, was bei besonders hohen Datenraten oder komplexen Berechnungen sinnvoll ist.

Eine Illustration beider Strategien ist in Abbildung 2 zu sehen. Links ist der Einsatz der

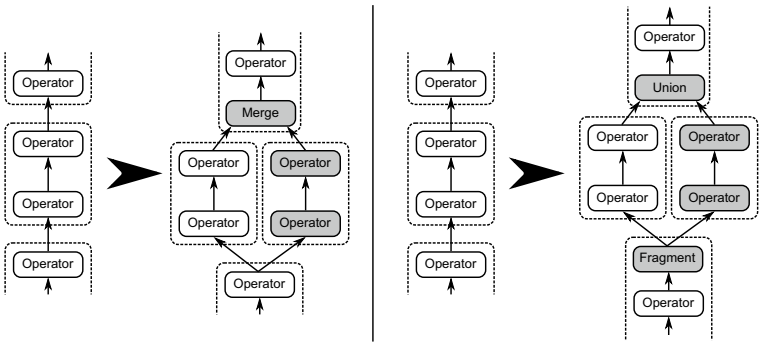


Abbildung 2: Verwendung der Replikations- (links) und der horizontalen Fragmentierungsstrategie (rechts).

Replikation als Modifikationsstrategie zu sehen: der Teilgraph wird kopiert und die replizierten (Teil-) Ergebnisse werden mittels eines speziellen Merge-Operator vereinigt. Der *Merge*-Operator erkennt und entfernt Duplikate in den Datenströmen, sodass die Replikation die Verarbeitungsergebnisse nicht unnötig vervielfacht. Im Rahmen der horizontalen Fragmentierung werden Teilgraphen ebenfalls kopiert (in der Abbildung rechts). Der vorgelagerte *Fragment*-Operator zerlegt den eintreffenden Datenstrom in disjunkte Fragmente, die parallel verarbeitet werden (bspw. mittels Hashwerten der Datenstromelemente). Der Union-Operator vereinigt die Teilmengen schließlich zu einem Datenstrom. In der Modifikationsphase ist ebenfalls vorgesehen, dass Entwickler eigene Strategien entwickeln und einsetzen (z.B. vertikale Fragmentierung).

3.3 Allokation

Die dritte Phase – die Allokation – beinhaltet die Aufgabe, Teilanfragen den Knoten im Netzwerk zur Ausführung zuzuordnen. Auch hier sind in Abhängigkeit zum vorliegenden Netzwerk verschiedene Vorgehensweisen vorstellbar, sodass im Rahmen dieser Arbeit mehrere Strategien betrachtet werden. Jede *Allokationsstrategie* erhält als Eingabe die Menge an (ggfs. replizierten und/oder fragmentierten) Teilgraphen. Die Ausgabe umfasst eine 1:n-Zuordnung zwischen ausführenden Knoten und Teilanfragen. Das bedeutet, dass ein Knoten mehrere Teilgraphen erhalten kann, jedoch wird jeder Teilgraph genau einem Knoten zugeordnet. Folgende Allokationsstrategien wurden bisher verfolgt:

Nutzerbasiert Der Nutzer gibt die Zuordnung vor (bspw. über eine grafische Oberfläche oder durch Annotationen im Anfragetext). Der Nutzer kann über Spezialwissen verfügen, die es ermöglichen, eine optimale Zuordnung anzugeben.

Round-Robin Die Teilgraphen werden der Reihe nach an die Knoten verteilt.

Lastorientiert Die Teilgraphen werden an die Knoten verteilt, welche aktuell die geringste Auslastung aufweisen. Durch diese Vorgehensweise kann die Systemlast im Netzwerk verteilt werden.

Contract-Net Für jeden Teilgraphen wird eine Auktion ausgeschrieben, und jeder Knoten kann bei Interesse ein Gebot abgeben. Ein Gebot beschreibt die Bereitschaft, den Teilgraphen zu übernehmen. Dabei kann ein Gebot aus verschiedenen Faktoren zusammengesetzt werden. Beispielsweise spielt die Menge an verfügbaren Systemressourcen eine Rolle: Je mehr Ressourcen frei sind, desto besser kann der Teilgraph ausgeführt werden. Der Knoten mit dem höchsten Gebot erhält schlussendlich den Teilgraphen.

Ist das Netzwerk bekannt und sind alle Knoten homogen, kann Round-Robin für eine schnelle und einfache Verteilung der Anfrage genutzt werden. Sollen Auslastungen der Knoten berücksichtigt werden, ist die lastorientierte Strategie vorzuziehen. Sie kann auch eingesetzt werden, wenn die Knoten über unterschiedliche Leistungskapazitäten verfügen. Contract-Net sollte benutzt werden, wenn die Autonomie der Knoten zu berücksichtigen ist (d.h., die Knoten entscheiden selbstständig, welche Teilgraphen sie ausführen wollen). Das in dieser Arbeit vorgestellte Konzept sieht vor, dass Entwickler eigene Allokationss-

trategien implementieren können, um bspw. domänenspezifisches Wissen einzusetzen oder spezielle Netzwerkstrukturen zu berücksichtigen.

4 Aktueller Stand

Das oben beschriebene Konzept wurde in Odysseus [AGG+12] als zusätzliche Komponente implementiert. Jede oben genannte Strategie ist verfügbar und kann in kontinuierlichen Anfragen eingesetzt werden. Dadurch kann Odysseus in unterschiedlichen Netzwerk-Architekturen eingesetzt werden, ohne dass umfangreiche Änderungen an der Verteilung durchgeführt werden müssen (es muss lediglich die Strategieauswahl angepasst werden). Im Folgenden werden zwei Anwendungsbeispiele von Odysseus vorgestellt, die aufzeigen sollen, wie das oben genannte Konzept flexibel eingesetzt werden kann.

Anwendungsfall 1: In einem Anwendungsfall wird Odysseus auf mehrere Knoten in einem heterogenen und autonomen P2P-Netzwerk eingesetzt, um Sportereignisse in Echtzeit auszuwerten. Die Daten werden mit Hilfe von aktiven Sensoren aufgenommen und an das Netzwerk gesendet. Die Sensoren sind dabei an spielrelevanten Entitäten wie den Spielern und dem Ball angebracht. Die Analyse geschieht mit zuvor verteilten kontinuierlichen Anfragen. Da zum einen ein solches Sensornetzwerk mehrere tausend Datenstromelemente pro Sekunde erzeugen kann und zum anderen die Analyse dieser Daten teuer ist, bietet sich eine Anfrageverteilung an, die in Listing 2 dargestellt ist. Konkret sollen

```
1 #NODE_PARTITION operatorsetcloud
2 #NODE_MODIFICATION fragmentation_horizontal hash n
3 #NODE_ALLOCATE contractnet
4
5 originale Analyse-Anfrage
```

Listing 2: Beispielhafte Verwendung der Anfrageverteilung für eine Sportanalyse.

Operatoren, die viel Last erzeugen, auf unterschiedlichen Knoten ausgeführt werden (*OperatorSetCloud*-Partitionierungsstrategie). Dadurch werden unterschiedliche Sportanalysen von verschiedenen Knoten des Netzwerks übernommen. Zusätzlich soll der Datenstrom fragmentiert werden, um die Systemlast für einzelne Knoten zu verringern (Modifikationsstrategie hash-basierte *horizontale Fragmentierung* und *n* Fragmenten). Da die Knoten heterogen und autonom sind, wird in diesem Fall die *Contract-Net*-Allokationsstrategie eingesetzt.

Anwendungsfall 2: In einem Windpark liefert jedes Windrad kontinuierlich Statusinformationen, die als Datenstrom interpretiert werden (z.B. die aktuell erzeugte Energie sowie Windrichtung und -geschwindigkeit). Diese Datenströme werden zur Überwachung und Kontrolle der Windräder benötigt und an ein homogenen Cluster aus Odysseus-Instanzen gesendet. Es können spezielle Datenstromelemente versendet werden, die Alarmmeldungen oder Störungen signalisieren. Aus diesem Grund ist es wichtig, dass jedes Datenstromelement (jede Alarmmeldung oder Störung) verarbeitet wird, auch wenn ein Knoten in dem verarbeitendem Cluster ausfällt. Listing 3 zeigt eine mögliche Anfrageverteilung für dieses

Szenario unter der Verwendung eines homogenen Clusters. Konkret sollen alle Operatoren

```
#NODE_PARTITION querycloud
#NODE_MODIFICATION replication n
#NODE_ALLOCATE roundrobin

originale Überwachungs-Anfrage
```

Listing 3: Beispielhafte Verwendung der Anfrageverteilung für eine Windpark-Überwachung.

der Anfrage auf einen Knoten ausgeführt werden, da die Cluster-Knoten mit ausreichend Ressourcen ausgestattet sind (*QueryCloud*-Partitionierungsstrategie). Zusätzlich wird der Datenstrom repliziert, um die Ausfallsicherheit zu erhöhen und um Alarmmeldungen nicht zu verlieren (Modifikationsstrategie *Replikation* mit *n* Replikaten). Als Allokator kommt die *Round-Robin*-Strategie zum Einsatz, da es sich um ein homogenes Cluster mit identischen Knoten handelt.

5 Zusammenfassung

Für eine Anfrageverteilung in verteilten DSMSs gibt es je nach Netzwerk-Architektur und Anwendungsfall unterschiedliche Strategien. Viele Systeme haben sich auf eine Verteilungsstrategie spezialisiert, wodurch sie nur mit großem Aufwand an eine Änderung der Netzwerk-Architektur angepasst werden können. Außerdem ist die Anfrageverteilung in vielen Systemen nicht durch den Nutzer konfigurierbar, was es unmöglich macht anwendungsspezifische Kenntnisse einzubringen.

Aus diesem Grund wurde in dieser Arbeit ein modularer Konzept für eine flexible und erweiterbare Anfrageverteilung in verteilten DSMSs vorgestellt. Das Konzept sieht dabei einen logischen Operatorgraphen als Eingabe vor und liefert verteilte Teilgraphen als Ausgabe. Strukturell umfasst er drei Schritte: (1) Partitionierung, (2) Modifikation und (3) Allokation.

Bei der Partitionierung wird der logische Operatorgraph in disjunkte Teilgraphen zerlegt, um die Operatoren zu identifizieren, die gemeinsam auf einem Knoten im Netzwerk ausgeführt werden sollen. Die optionale Modifikation erlaubt es Mechanismen wie Fragmentierung oder Replikation zu verwenden, indem die Teilgraphen modifiziert werden. In der Allokationsphase werden die einzelnen (modifizierten) Teilgraphen Knoten im Netzwerk zugewiesen.

Für jeden der drei Schritte gibt es Schnittstellen, wodurch unabhängige Strategien miteinander kombiniert werden können. Dieser modulare Aufbau ermöglicht zum einen eine individuelle Anfrageverteilung. Zum anderen können bereits vorhandene Strategien aus anderen Arbeiten und Systemen (z.B. eine Allokationsstrategie) integriert werden. In dieser Arbeit wurden für jeden der drei Teilschritte beispielhafte Strategien vorgestellt.

Das Konzept wurde im DSMS Odysseus als zusätzliche Komponente implementiert und erfolgreich in verschiedenen Anwendungsszenarien eingesetzt. Zwei Anwendungsszena-

rien wurden in dieser Arbeit kurz vorgestellt: (1) Die Sportanalyse in Echtzeit mittels einem P2P-Netzwerk aus heterogenen Knoten und (2) die Überwachung eines Windparks. Odysseus musste in beiden Anwendungsfällen lediglich bei der Strategieauswahl angepasst werden. Dies zeigt, dass das oben beschriebene Konzept zur Verteilung kontinuierlicher Anfragen flexibel und erweiterbar ist.

Literatur

[AGG+12] H.-Jürgen Appelrath, Dennis Geesen, Marco Grawunder, Timo Michelsen und Daniela Nicklas. Odysseus: a highly customizable framework for creating efficient event stream management systems. DEBS '12, Seiten 367–368. ACM, 2012.

[BMS+13] Aydin Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders und Christian Schulz. Recent Advances in Graph Partitioning. *CoRR*, abs/1311.3144, 2013.

[CBB+03] Mitch Cherniack, Hari Balakrishnan, Magdalena Balazinska, Donald Carney, Ugur Cetintemel, Ying Xing und Stan Zdonik. Scalable Distributed Stream Processing. In *CIDR 2003 - First Biennial Conference on Innovative Data Systems Research*, Asilomar, CA, January 2003.

[DLBMW11] Michael Daum, Frank Lauterwald, Philipp Baumgärtel und Klaus Meyer-Wegener. Kalibrierung von Kostenmodellen für föderierte DSMS. In *BTW Workshops*, Seiten 13–22, 2011.

[GAW09] Buğra Gedik, Henrique Andrade und Kun-Lung Wu. A Code Generation Approach to Optimizing High-performance Distributed Data Stream Processing. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, Seiten 847–856, New York, NY, USA, 2009. ACM.

[GJPPM+12] Vincenzo Gulisano, Ricardo Jimenez-Peris, Marta Patino-Martinez, Claudio Soriente und Patrick Valduriez. StreamCloud: An Elastic and Scalable Data Streaming System. *IEEE Transactions on Parallel and Distributed Systems*, 23(12):2351–2365, 2012.

[KSKR05] Richard Kuntschke, Bernhard Stegmaier, Alfons Kemper und Angelika Reiser. Streamglobe: Processing and sharing data streams in grid-based p2p infrastructures. In *Proceedings of the 31st international conference on Very large data bases*, Seiten 1259–1262. VLDB Endowment, 2005.

[LHKK12] Simon Loesing, Martin Hentschel, Tim Kraska und Donald Kossmann. Stormy: an elastic and highly available streaming service in the cloud. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, EDBT-ICDT '12, Seiten 55–60, New York, NY, USA, 2012. ACM.

[TTS+14] Ankit Toshniwal, Siddarth Taneja, Amit Shukla, Karthikeyan Ramasamy, Jignesh M. Patel, Sanjeev Kulkarni, Jason Jackson, Krishna Gade, Maosong Fu, Jake Donham, Nikunj Bhagat, Sailesh Mittal und Dmitriy V. Ryaboy. Storm@twitter. In *SIGMOD Conference*, Seiten 147–156, 2014.

[WK09] Daniel Warneke und Odej Kao. Nephele: Efficient Parallel Data Processing in the Cloud. In *Proceedings of the 2Nd Workshop on Many-Task Computing on Grids and Supercomputers*, MTAGS '09, Seiten 8:1–8:10, New York, NY, USA, 2009. ACM.

Placement-Safe Operator-Graph Changes in Distributed Heterogeneous Data Stream Systems

Niko Pollner, Christian Steudtner, Klaus Meyer-Wegener
Computer Science 6 (Data Management)
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
niko.pollner@fau.de, c.steudtner@gmx.net, klaus.meyer-wegener@fau.de

Abstract: Data stream processing systems enable querying continuous data without first storing it. Data stream queries may combine data from distributed data sources like different sensors in an environmental sensing application. This suggests distributed query processing. Thus the amount of transferred data can be reduced and more processing resources are available.

However, distributed query processing on probably heterogeneous platforms complicates query optimization. This article investigates query optimization through operator graph changes and its interaction with operator placement on heterogeneous distributed systems. Pre-distribution operator graph changes may prevent certain operator placements. Thereby the resource consumption of the query execution may unexpectedly increase. Based on the operator placement problem modeled as a task assignment problem (TAP), we prove that it is NP-hard to decide in general whether an arbitrary operator graph change may negatively influence the best possible TAP solution. We present conditions for several specific operator graph changes that guarantee to preserve the best possible TAP solution.

1 Introduction

Data stream processing is a well suited technique for efficient analysis of streaming data. Possible application scenarios include queries on business data, the (pre-)processing of measurements gathered by environmental sensors or by logging computer network usage or online-services usage.

In such scenarios data often originate from distributed sources. Systems based on different software and hardware platforms acquire the data. With distributed data acquisition, it is feasible to distribute query processing as well, instead of first sending all data to a central place. Some query operators can be placed directly on or near the data acquisition systems. This omits unnecessary transfer of data that are not needed to answer the queries, and partitions the processing effort. Thus querying high frequency or high volume data becomes possible that would otherwise require expensive hardware or could not be processed at all. Also data acquisition devices like wireless sensor nodes profit from early operator execution. They can save energy if data is filtered directly at the source.

Problem Statement Optimization of data stream queries for a distributed heterogeneous execution environment poses several challenges: The optimizer must decide for each operator on which processor it should be placed. Here and in the remainder of this article, the term processor stands for a system that is capable to execute operators on a data stream. A cost model is a generic base for the operator placement decision. It can be adapted to represent the requirements of specific application scenarios, so that minimal cost represents the best possible operator distribution. Resource restrictions on the processors and network links between them must also be considered. In a heterogeneous environment costs and capacities will vary among the available processors. The optimizer can optimize the query graph before and after the placement decision. Pre-placement changes of the query graph may however foil certain placements and a specific placement limits the possible post-placement algebraic optimization. For example, a change of the order of two operators can increase costs if the first operator of the original query was available directly on the data source and the now first operator in the changed query is not. This must be considered when using common rules and heuristics for query graph optimization.

Contribution We investigate the influence of common algebraic optimization techniques onto a following operator placement that is modeled as a task assignment problem (TAP). We prove that the general decision whether a certain change of the query graph worsens the best possible TAP solution is NP-hard. We then present analysis of different common operator graph changes and state the conditions under which they guarantee not to harm the best possible placement. We do not study any special operator placement algorithm, but focus on preconditions for graph changes.

Article Organization The following section gives an overview on related work from both the fields of classical database query optimization and data stream query optimization. Sect. 3 introduces the TAP model for the operator placement. It is the basis for the following sections. We prove the NP-hardness of the query-graph-change influence in Sect. 4 and present the preconditions for special graph changes in Sect. 5. The next section shows how to use the preconditions with an exemplary cost model for a realistic query. In the last section we conclude and present some ideas for further research.

2 Related Work

This section presents related work on operator graph optimization from the domains of data base systems (DBS) and data stream systems (DSS). Due to space limitations, we are unfortunately only able to give a very rough overview.

Query optimization in central [JK84] as well as in distributed [Kos00] DBS is a well studied field. Basic ideas like operator reordering are also applicable to DSS. Some operators, especially blocking operators, however have different semantics. Other techniques like optimization of data access have no direct match in DSS. Strict resource restrictions are

also rarely considered with distributed DBS because they are not thought to run on highly restricted systems.

The authors of [HSS+14] present a catalog of data stream query optimizations. For each optimization, realistic examples, preconditions, its profitability, and dynamic variants are listed. Among operator graph changes the article also presents other optimizations, like load-shedding, state sharing, operator placement and more. They impose the question for future research, in which order different optimizations should be performed. In the paper at hand, we go a first step into this direction by studying the influence of operator graph changes to following placement decisions. We detail the impact of all the five operator graph changes from [HSS+14]. We think that these changes cover the common query graph optimizations.

The articles [TD03] and [NWL+13] present different approaches to dynamic query optimization. The basic idea is that the order in which tuples visit operators is dynamically changed at runtime. The concept of distributed Eddies from [TD03] decides this on a per tuple basis. It does not take the placement of operators into account. Query Mesh [NWL+13] precreates different routing plans and decides at runtime which plan to use for a set of tuples. It does not consider distributed query processing.

3 Operator Placement as Task Assignment Problem

The operator placement can be modeled as a TAP. Operators are represented as individual tasks. We use the following TAP definition, based on the definition in [DLB+11].

P is the set of all query processors. L is the set of all communication channels. A single communication channel $l \in L$ is defined as $l \subseteq \{P \times P\}$. A communication channel subsumes the communication between processors that share a common medium.

T is the set of all operators. The data rate (in Byte) between two operators is given by $r_{t_1 t_2}, t_1, t_2 \in T$. The operators and rates represent the query graph.

c_{tp} are the processing costs of operator t on processor p. $k_{p_1 p_2}$ gives the cost of sending one Byte of data between processor p_1 and processor p_2. The costs are based on some cost model according to the optimization goal. Since cost models are highly system and application specific, we do not assume a specific cost model for our research of query-graph-change effects. Sect. 6 shows how to apply our findings to an exemplary cost model. [Dau11, 98–121] presents methods for the estimation of operator costs and data rates.

The distribution algorithm tries to minimize the overall cost. It does this by minimizing term (1), considering the constraints (2) - (5). The sought variables are x_{tp}. $x_{tp} = 1$ means that task t is executed on processor p. The first sum in equation (1) are the overall processing costs. The second sum are all communication costs.

$$\min \sum_{t \in T} \sum_{p \in P} c_{tp} x_{tp} + \sum_{t_1 \in T} \sum_{p_1 \in P} \sum_{t_2 \in T} \sum_{p_2 \in P} k_{p_1 p_2} r_{t_1 t_2} x_{t_1 p_1} x_{t_2 p_2} \tag{1}$$

$$\sum_{t \in T} c_{tp} x_{tp} \leq b(p), \forall p \in P \quad (2)$$

$$\sum_{(p_1,p_2) \in l} \sum_{t_1 \in T} \sum_{t_2 \in T} r_{t_1 t_2} x_{t_1 p_1} x_{t_2 p_2} \leq d(l), \forall l \in L \quad (3)$$

$$\sum_{p \in P} x_{tp} = 1, \forall t \in T \quad (4)$$

$$x_{tp} \in \{0, 1\}, \forall p \in P, \forall t \in T \quad (5)$$

Constraint (2) limits the tuple processing cost of the operators on one processor to its capacity $b(p)$. Constraint (3) limits the communication rate on one communication channel to its capacity $d(l)$. Constraints (4) and (5) make sure that each task is distributed to exactly one single processor.

Our findings are solely based on the objective function together with the constraints. We do not assume any knowledge about the actual distribution algorithm. There exist different heuristics for solving a TAP. See e.g. [DLB+11] and [Lo88].

4 Generic Operator-Graph-Change Influence Decision

A single algebraic query transformation changes the TAP in numerous ways. For example an operator reordering changes multiple data rates, which are part of multiple equations inside the TAP. When some of those factors increase, it is hard to tell how it affects a following operator placement. The transformed query might even become impossible to execute.

One way to determine the usefulness of a given transformation is to compare the minimum costs of both the original and the transformed query graph. If the transformed query graph has lower or equal cost for the optimal operator placement, i.e. lower or equal minimum cost, than the original query, the transformation has a non-negative effect. $U(Q)$ denotes the query graph that results from applying a change U to the original query Q. Since the operator placement needs to solve a TAP, an NP-complete problem, it is not efficient to compute the placement for each possible transformation. A function `CompareQuery(`Q`, `$U(Q)$`)`, that compares two queries and returns true iff $U(Q)$ has smaller or equal minimal costs than Q would solve the problem.

Sentence. `CompareQuery(`Q`, `$U(Q)$`)` *is NP-hard.*

Definition. *U_{tp} is a transformation that allows task t only to be performed by processor p. All other aspects of $U_{tp}(Q)$ are identical to Q. Both Q and $U_{tp}(Q)$ have equal costs when operators are placed in the same way, i.e. as long as t is placed on p.*

Proof. Given `CompareQuery(`Q`, `$U(Q)$`)` and transformations U_{tp} it is possible to compute the optimal distribution. For each task t it is possible to compare Q and $U_{tp}(Q)$ for

each processor p. If `CompareQuery(`Q`,` $U_{tp}(Q)$`)` returns true $U_{tp}(Q)$ has the same minimum cost as Q. Thus the optimal placement of t is p. The algorithm in pseudo code:

```
ComputeDistribution(Q) {
  foreach (t in Tasks) {
    foreach (p in Processors) {
      if (CompareQuery(Q, U_tp(Q)) == true) {
        DistributeTaskProcessor(t, p);
          // makes sure t will be distributed to p
        break; // needed if multiple distributions exist
} } } }
```

`ComputeDistribution(`Q`)` calls `CompareQuery(`Q`,` $U(Q)$`)` at most $|T| \cdot |P|$ times. This is a polynomial time reduction of `ComputeDistribution(`Q`)`. To compute the optimal distribution it is necessary to solve the TAP, an NP-complete problem. This proves that `CompareQuery(`Q`,` $U(Q)$`)` is NP-hard. □

5 Specific Query Graph Changes

While the general determination of a query graph change's impact is NP-hard, it can easily be determined for specific cases. If a transformation neither increases variables used for the TAP nor adds new variables to the TAP, it is trivial to see that all valid operator placement schemes are still valid after the transformation. For the transformed query exist operator placements with lower or equal costs than the original query's costs: the original optimal placement is still valid and has lesser or equal costs.

We establish preconditions for all the five operator graph changes from [HSS+14]. If the preconditions are met the transformation is safe. That means for each valid operator placement scheme of the original query exists a valid scheme for the transformed query with equal costs. So the preconditions especially guarantee that the minimum costs do not rise. However, if heuristic algorithms are used for solving the TAP, they may fail to find an equally good solution for the transformed query as they did for the original query and vice versa, because local minima may change.

Table 1 shows all preconditions at a glance. We now justify why these preconditions hold.

Notation Most of the used notation directly follows from the TAP, especially c_{tp} and $r_{t_1 t_2}$. The cost c_{Ap} of an operator A on the processor p depends on the input stream of A and thus on the overall query executed before that operator. Query graph changes affect the input streams of operators an thus also change the costs needed to execute those operators. In order to distinguish between the original and the changed query we use $U(A)$ to indicate the operator A with the applied query change. $U(A)$ and A behave in the same way, but may have different cost, since they work on different input streams. The costs $c_{U(A)p}$ are needed to execute $U(A)$ on p and the following operator t receives an input stream with

the data rate $r_{U(A)t}$. In addition r_I denotes the input data stream and r_O denotes the output data stream.

Operator Reordering Operator reordering switches the order of two consecutive operators. The operator sequence $A \rightarrow B$ is transformed to $U(B) \rightarrow U(A)$. In the original query, operator A is placed on processor p_A and operator B on p_B. It is possible that p_A is the same processor as p_B, but it is not known whether both operators are on the same processor, so this cannot be assumed. $p_A = p_B$ would result in a set of preconditions that are easier to fulfill than the preconditions we present. The transformed query can place the operators $U(B)$ and $U(A)$ on any of the processors p_A and p_B.

Case 1: $U(B)$ is placed on p_A and $U(A)$ is placed on p_B. To insure the validity of all distributions, the transformed operators' cost must not exceed the cost of the other original operator, which results in equations (6) and (7). Since the reordering affects the data rates between operators, precondition (8) must hold.

Case 2: Both operators are placed on p_A. This adds an internal communication inside p_A to the operator graph. Equation (9) ensures that internal communication is not factored into the TAP constraints and cost function. The sum of the cost for both transformed operators must be smaller or equal than the cost of A, which is described by equation (10). The changed data rates are reflected in equation (11).

Case 3: Both operators are placed on p_B. This case is similar to case 2 and can be fulfilled with the preconditions given by equations (9), (12) and (13).

Case 4: The remaining option, $U(B)$ is placed on p_B and $U(A)$ is placed on p_A, can be viewed as changed routing. Since the remaining distribution of the query is unknown, the changed routing can be problematic and this option is inherently not safe. It is possible that p_A processes the operator that sends the input to A and that p_B has an operator that processes the output stream of B. In this situation the changed routing causes increased communication cost, since the tuples must be send from p_A to p_B (applying B) to p_A (applying A) to p_B instead of only sending them once from p_A to p_B.

If one of the operators has more than one input stream not all cases can be used. Even if the stream does not need to be duplicated, if A has additional input streams only case 2 is valid. The other cases are not safe anymore, because the transformation changes the routing of the second stream from destination p_A to destination p_B. Similar, if B has additional input streams only case 3 is safe.

Redundancy Elimination This query change eliminates a redundant operator: the query graph has an operator A on two different positions processing the same input stream, duplicated by another operator. This change works by removing one of the instances of A and duplicating its output.

The original query consist of three operators. Operator D (Dup Split in [HSS+14]) is placed on p_D, while an instance of A is placed both on p_1 and p_2. The transformed query

Transformation	Case	Precondition ($\forall p \in P$)	
Operator reordering	Case 1: $U(B)$ on p_A $U(A)$ on p_B	$c_{Ap} \geq c_{U(B)p}$	(6)
		$c_{Bp} \geq c_{U(A)p}$	(7)
		$r_{AB} \geq r_{U(B)U(A)}$	(8)
	Case 2: $U(B)$ on p_A $U(A)$ on p_A	$k_{pp} = 0 \wedge \forall l \in L : (p,p) \notin l$	(9)
		$c_{Ap} \geq c_{U(B)p} + c_{U(A)p}$	(10)
		$r_{AB} \geq r_O$	(11)
	Case 3: $U(B)$ on p_B $U(A)$ on p_B	$k_{pp} = 0 \wedge \forall l \in L : (p,p) \notin l$	(9)
		$c_{Bp} \geq c_{U(B)p} + c_{U(A)p}$	(12)
		$r_{AB} \geq r_I$	(13)
Redundancy elimination	–	$k_{pp} = 0 \wedge \forall l \in L : (p,p) \notin l$	(9)
		$c_{Dp} \geq c_{U(A)p} + c_{U(D)p}$	(14)
Operator separation	–	$k_{pp} = 0 \wedge \forall l \in L : (p,p) \notin l$	(9)
		$c_{Ap} \geq c_{A_1p} + c_{A_2p}$	(15)
Fusion	Case 1: All on p_A	$c_{Ap} \geq c_{Cp}$	(16)
		$r_{AB} \geq r_O$	(17)
	Case 2: All on p_B	$c_{Bp} \geq c_{Cp}$	(18)
		$r_{AB} \geq r_I$	(19)
Fission	–	$k_{pp} = 0 \wedge \forall l \in L : (p,p) \notin l$	(9)
		$c_{Ap} \geq c_{Sp} + c_{Mp} + \sum_{U(A)} c_{U(A)p}$	(20)

Table 1: Preconditions for safe query graph changes that must be fulfilled for all processors. If an operator is not available on some processors, the preconditions can be assumed fulfilled for these processors. It is sufficient that the preconditions of one case are fulfilled.

consists of the operators $U(A)$ and $U(D)$, with $U(D)$ duplicating the output instead of the input. The only possibility to place the transformed query without changing routing is to place both $U(A)$ and $U(D)$ on p_D. The additional internal communication, due to the additional operator on p_D, again forces equation (9). To ensure that any processor can perform the transformed operators, equation (14) is necessary.

In some situations (when p_D is the same processor as p_1 or p_2) the change is safe as long as A does not increase the data rate. But since it is unknown how the operators will be placed, this requirement is not sufficient.

Operator Separation The operator separation splits an operator A into the two operators $A_1 \rightarrow A_2$. Additional internal communication results in precondition (9). Equation (15) ensures that the separated operators' costs are together less than or equal to A's cost.

Fusion Fusion is the opposite transformation to operator separation. The two operators $A \rightarrow B$ are combined to the single operator C (a superbox in [HSS+14]). For the original query A is placed on p_A and B on p_B. The combined operator can be placed on either p_A or p_B. The cost for C must not exceed the cost of p_A or p_B respectively. In addition, the data rates are affected and thus also add preconditions. So either the fulfillment of equations (16) and (17) (if C is placed on p_A) or (18) and (19) (if C is placed on p_B) guarantee the safety of this change. A special case of the fusion is the elimination of an unneeded operator, i.e. removing the operator does not change the query result. Since the redundant operator can change the data rate of a stream (e.g. a filter applied before a more restrictive filter) it still needs to fulfill the preconditions to be safe.

Fission The original query is only the single operator A. Fission replaces A by a partitioned version of it, by applying a split operator S, multiple versions of $U(A)$, which can potentially be distributed across different processors, and finally a merge operator M to unify the streams again. Since it is unknown whether other processors exist that can share the workload profitably, the transformed operators must be placed on the processor that executed the original A. This is safe when preconditions (9) and (20) hold. These equations demand that the combined costs of the split, merge and all parallel versions of $U(A)$ can be executed by all processors with smaller or equal cost than the original A.

6 Application

Given a query and a DSS it is now possible to test whether a specific change is safe. Using an exemplary cost model we examine a simple example query.

Cost Model [Dau11, 91–98] presents a cost model that will be used for the following example. We use a filter and a map operator, which have the following costs:

$$C_{Filter} = \lambda_i C_{Fil} + \lambda_o C_{AppendOut} \tag{21}$$
$$C_{Map} = \lambda_i C_{proj} + \lambda_o C_{AppendOut} \tag{22}$$

C_{Fil} and C_{proj} are the costs associated with filtering respectively projecting an input tuple arriving at the operator. $C_{AppendOut}$ represents the costs of appending one tuple to the output stream. λ_i is the input stream tuple rate, while λ_o is the output stream tuple rate. For those operators λ_o is proportional to λ_i and the equations (21) and (22) can be simplified to $\lambda_i f_{Op}$, where f_{Op} is the cost factor of operator O on processor p for one tuple.

Using these simplified equations, the assumption that the tuple rate is proportional to the data rate and costs and selectivities are non-zero, equations (6) to (8) can be rewritten as:

$$\lambda_I f_{Ap} \geq \lambda_I f_{Bp} \quad \Leftrightarrow \quad \frac{f_{Ap}}{f_{Bp}} \geq 1 \tag{23}$$
$$\sigma_A \lambda_I f_{Bp} \geq \sigma_{U(B)} \lambda_I f_{Ap} \quad \Leftrightarrow \quad \frac{\sigma_A}{\sigma_{U(B)}} \geq \frac{f_{Ap}}{f_{Bp}} \tag{24}$$
$$\sigma_A \lambda_I \geq \sigma_{U(B)} \lambda_I \quad \Leftrightarrow \quad \frac{\sigma_A}{\sigma_{U(B)}} \geq 1 \tag{25}$$

The equations for the other two cases shown in table 1 can be similarly rewritten. Equations (23) to (25) show that there are relatively few values to compare: We need the ratio of the operator selectivity and for each processor the ratio of operator costs.

Example We examine the simple query of a map operator M followed by a filter F applied on a stream containing image data monitoring conveyor belts transporting freshly produced items. The query supports judging the quality of the current production run. Operator M classifies each tuple (and thus each observed produced item) into one of several quality classes and is rather expensive. F filters the stream for one conveyor belt, because different conveyor belts transport different items and are observed by different queries.

M does not change the data rate of the stream. It simply replaces the value *unclassified* already stored inside the input stream for each tuple with the correct classification and thus has a selectivity of 1. There are multiple types of processors available inside the production hall. Depending on the processor type the ratio $\frac{f_{Mp}}{f_{Fp}}$ differs quite a bit, but overall M is more expensive: this ratio fluctuates between 2 and 10. Equations (23) to (25) show that the selection push down is always safe if $\sigma_{U(F)}$ is smaller or equal than 0.1: In this case it is always possible that the two operators switch their places without violating additional constraints of the TAP. If $\sigma_{U(F)}$ is greater than 0.1 this change is not necessarily safe. It is possible that the preconditions of one of the other two cases (both operators on the same processor) are fulfilled or another good distribution is possible, but the latter cannot be tested in a reasonable time as we discussed in Sect. 4.

7 Conclusion

We presented our findings on the interaction between optimization through query graph changes and the placement of operators on different heterogeneous processing systems. We first motivated our research and defined the problem. Existing work on query optimization through operator graph changes in the context of DMS and DSS was presented, none of which studied the interaction with operator placement. The next section presented the TAP model of the distribution problem. We showed that it is NP-hard to decide in general if an arbitrary query graph change can negatively influence the best possible operator placement scheme. Based on a selection of common query graph changes from the literature, we deduced preconditions under which operator placement does not mind the changes. The last section showed the application of our findings with an exemplary cost model for a realistic query.

The preconditions for safe operator graph changes are quite restrictive. They severely limit the possible changes if followed strictly. As with general query optimization, development of heuristics to loosen certain preconditions seems promising. The preconditions presented in this article are the basis for such future work. Another interesting field is the direct integration of query graph optimization in the usually heuristic distribution algorithms. Distribution algorithms could be extended to consider query graph changes in addition to the operator placement. We plan to investigate these ideas in our future research.

References

[Dau11] M. Daum. *Verteilung globaler Anfragen auf heterogene Stromverarbeitungssysteme*. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2011.

[DLB+11] M. Daum, F. Lauterwald, P. Baumgärtel, N. Pollner, and K. Meyer-Wegener. Efficient and Cost-aware Operator Placement in Heterogeneous Stream-Processing Environments. In *Proceedings of the 5th ACM International Conference on Distributed Event-Based Systems (DEBS)*, pages 393–394, New York, NY, USA, 2011. ACM.

[HSS+14] M. Hirzel, R. Soulé, S. Schneider, B. Gedik, and R. Grimm. A Catalog of Stream Processing Optimizations. *ACM Comput. Surv.*, 46(4):1–34, 2014.

[JK84] M. Jarke and J. Koch. Query Optimization in Database Systems. *ACM Comput. Surv.*, 16(2):111–152, 1984.

[Kos00] D. Kossmann. The State of the Art in Distributed Query Processing. *ACM Comput. Surv.*, 32(4):422–469, 2000.

[Lo88] V. M. Lo. Heuristic Algorithms for Task Assignment in Distributed Systems. *IEEE Transactions on Computers*, 37(11):1384–1397, 1988.

[NWL+13] R. V. Nehme, K. Works, C. Lei, E. A. Rundensteiner, and E. Bertino. Multi-route Query Processing and Optimization. *J. Comput. System Sci.*, 79(3):312–329, 2013.

[TD03] F. Tian and D. J. DeWitt. Tuple Routing Strategies for Distributed Eddies. In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29*, VLDB '03, pages 333–344. VLDB Endowment, 2003.

Herakles: A System for Sensor-Based Live Sport Analysis using Private Peer-to-Peer Networks

Michael Brand, Tobias Brandt, Carsten Cordes, Marc Wilken, Timo Michelsen
University of Oldenburg, Dept. of Computer Science
Escherweg 2, 26129 Oldenburg, Germany
{michael.brand,tobias.brandt,carsten.cordes,marc.wilken,timo.michelsen}@uni-ol.de

Abstract: Tactical decisions characterize team sports like soccer or basketball profoundly. Analyses of training sessions and matches (e.g., mileage or pass completion rate of a player) form more and more a crucial base for those tactical decisions. Most of the analyses are video-based, resulting in high operating expenses. Additionally, a highly specialized system with a huge amount of system resources like processors and memory is needed. Typically, analysts present the results of the video data in time-outs (e.g., in the half-time break of a soccer match). Therefore, coaches are not able to view statistics during the match.

In this paper we propose the concepts and current state of Herakles, a system for live sport analysis which uses streaming sensor data and a Peer-to-Peer network of conventional and low-cost private machines. Since sensor data is typically of high volume and velocity, we use a distributed data stream management system (DSMS).

The results of the data stream processing are intended for coaches. Therefore, the front-end of Herakles is an application for mobile devices (like smartphones or tablets). Each device is connected with the distributed DSMS, retrieves updates of the results and presents them in real-time. Therefore, Herakles enables the coach to analyze his team during the match and to react immediately with tactical decisions.

1 Introduction

Sport is a highly discussed topic around the world. Fans, the media, teams and athletes discuss about performance, tactical decisions and mistakes. Statistics about sports are an important part of these discussions. Some statistics are calculated manually by people outside the playing field (e.g., counting ball contacts). Other statistics are retrieved automatically by computer-based analysis systems. However, such systems require expensive computers to calculate the statistics due to the huge amount of data to process. Not every team is able to buy and maintain them. Additionally, many systems need too much time to calculate the statistics, making it impossible to deliver them during the game. Many already existing sport analysis systems are using several cameras placed around the playing field. This increases the costs for buying and maintaining the system even further.

In this paper, we propose *Herakles*, a live sport analysis system based solely on conventional low-cost hardware and software we are currently working on. Instead of using costly cameras, Herakles uses a network of sensors attached to all game-relevant objects, e.g., the players and the ball. Since sensor systems are usually not allowed during games and

opponents won't wear the sensors anyway, Herakles focuses on the use for tryouts. The active sensors regularly send their actual position, acceleration and other values as *data streams*. Therefore, Herakles uses a distributed *data stream management system* (DSMS). With such a DSMS, it is possible to compute relevant sport statistics in real-time. In our work, real-time means "near to the actual event"(rather than in timeouts or after a match). But using a single machine with a DSMS installed invokes the possibility of single-point-of-failures: (1) a single DSMS instance would run the risk of overload and (2) the live analysis stops completely if the single DSMS instance fails. Therefore, Herakles uses multiple machines to share the processing load and to increase the reliability of the entire system. Currently, many already existing distributed DSMSs are using computer clusters, grids, etc. (e.g., [vdGFW+11]).

Since Herakles focuses on using a collection of smaller low-cost private machines like notebooks, a *private Peer-to-Peer* (P2P) network is more feasible. In our work, each peer is considered to be heterogeneous and autonomous. This makes it possible to use these types of private machines for distributed data stream processing of sensor data for live sport analysis. Although the peers are low-cost private machines, the system costs depend on the used sensor network.

To show the calculated sport statistics in a convenient way for different users, Herakles focuses on mobile devices for the presentation (e.g., smartphones or tablets). With this, the statistics can be directly shown in real-time. Additionally, Herakles abstracts from specific sensors as well as from a specific sport. Statistics for different sports can be calculated using different sensors but based on the same system. Subsequently, these statistics have to be presented in a way that the technological background is encapsulated from the user, e.g., the user does not need to configure the P2P network.

The remainder of this paper is structured as follows: Section 2 gives an overview of related sport analysis systems. The concept of Herakles is explained in Section 3. In Section 4 we give an overview on the current state of our implementation. Section 5 gives a prospect of future work and Section 6 concludes this paper.

2 Related Work

There are many existing sport analysis systems developed and used. However, the majority of them is built on cameras. Many of them are commercial products like Synergy Sports Technology[1] or Keemotion[2]. Other projects are scientific. For instance, Baum [Bau00] provides a video-based system for performance analysis of players and objects. It uses high-speed video cameras and was tested in the context of baseball. As mentioned earlier, there are many problems related to video-based systems, especially the high acquisition and maintenance costs.

Leo et al. [LMS+08] refer to a video-based soccer analysis of players and objects. In their approach, the ball and players are tracked by six video cameras. Then, the video data is

[1] http://corp.synergysportstech.com/
[2] http://www.keemotion.com/

received by six machines and processed by a central supervisor. The arising problems of this approach are the same as for Baum [Bau00].

Smeaton et al. [SDK+08] propose an analysis of the overall health of the sportsperson in the context of football games. Their sensor-based approach includes body sensors in combination with video recordings. The location is tracked by GPS, but the data can only be reviewed after the game and their approach currently handles only one person at a time.

Von der Grün [vdGFW+11] proposes a sensor-based approach called RedFIR. It uses expensive hardware for the real-time analysis (e.g., SAP HANA for a german soccer team[3]), which leads to high acquisition and maintenance costs like in [Bau00]. Additionally, their data is processed in an in-memory database instead of a DSMS.

To our best knowledge, there is no sport analysis system that uses a network of conventional, low-cost machines for processing streaming sensor data and that shows the results on a mobile device in real-time.

3 Concept

In this section, we give an overview of Herakles. Its architecture is shown in figure 1 and is divided into three components: (1) sensor network, (2) P2P network of DSMSs and (3) mobile device.

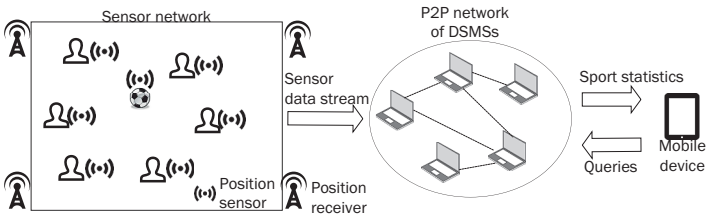


Figure 1: Architecture of Herakles with sensors, a P2P network and a mobile device.

We equip important game entities with sensors to form the sensor network. Each of these sensors sends the position and other information like speed and acceleration of the respective entity. This information is captured by receivers, which are placed around the playing field. These receivers are connected to the next component, the P2P network of conventional (private) machines with DSMSs installed.

The DSMSs receive the sensor data stream. To be independent from a specific sport and sensor technology, Herakles separates the incoming raw sensor data into multiple intermediate schemata at first (data abstraction). The DSMSs use continuous queries built on top of these intermediate schemata to continuously analyze the data and to calculate the sport statistics. With this, we can replace the sensors, but we do not need to change the continuous queries.

[3] http://tinyurl.com/o82mub5

The query results are streamed to the mobile device, the last component in the architecture of Herakles. The mobile device visualizes the statistics and the user can decide at any time, which statistics should be shown at the moment.

Inside Herakles, each statistic is mapped to a continuous query, which is distributed, installed and executed in the P2P network. The distribution is necessary to make sure that not a single peer processes a complete query (avoiding overloads).

In the following sections, we give a more detailed view about the components of Herakles. In section 3.1, we describe our data abstraction. The used P2P network and the distributed DSMS with it's features is explained in section 3.2. Finally, the presentation on mobile devices is explained in section 3.3.

3.1 Data Abstraction

To improve reusability and flexibility, Herakles separates the sensor data into different layers. In doing so, we can avoid direct dependencies between calculated statistics and a sensor's technology and data format. An overview of Herakles' data abstraction is shown in figure 2.

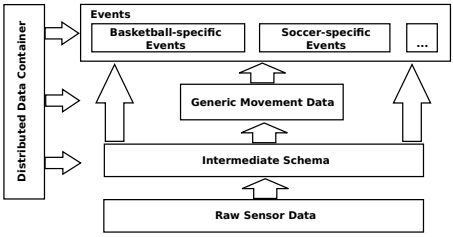


Figure 2: Data abstraction layers ranging from sensor schema up to sport specific schemata.

The sensor network sends its data as *raw sensor data*. Inside the P2P network, this data is converted into a common *intermediate schema* for standardization. This includes for example a unit conversion.

The intermediate schema improves the interchangeability in Herakles: While the sensor's data format can change, the intermediate schema stays unchanged. When different sensors are used, only the adapter between raw sensor data and the intermediate schema has to be rewritten. Other parts of Herakles including the continuous queries for the sport statistics stay unchanged.

In our opinion, generic movement information is interesting for more than one sport and should be reused. Therefore, on top of the intermediate schema, we differentiate between *generic movement data* (e.g., the player's position and running speed) and *sport-specific events* (e.g., shots on goal for soccer). Sport-specific events need to be created for each sport individually (e.g., basketball and soccer). By separating generic movement data from sport-specific events, we can reuse parts of continuous queries in different sports. In ge-

neral, Herakles uses continuous queries built on each other to transform the data: one continuous query receives the raw sensor data, transforms it into the intermediate schema and provides the results as an artificial data source. This source can be reused from other continuous queries to determine the movement data, identifying sport-specific events etc.

Another aspect of the data abstraction is the storage and the access to static values. For instance, the positions of the soccer-goals are needed in many statistics (e.g., for counting shots on goal or identifying goals itself). Another example is the identification, which sensor is attached to which player or ball. Since Herakles uses a decentralized and dynamic P2P network, a central server is not applicable. Additionally, it is not feasible to define the static values on each peer individually. Therefore, Herakles needs a decentralized way to share these static information across the network.

The data abstraction of Herakles uses a so-called *distributed data container* (DDC). It reads information (e.g., from a file) and distributes them automatically to other peers in the network without further configuration. Then, each peer can use these values inside its own DSMS. Changes at one peer in the network are propagated to the other peers automatically. With this, the user has to define the static values only at one peer.

3.2 Data Stream Management System

Herakles uses a distributed DSMS in a P2P network for processing the sensor data. Since the peers are private machines (e.g., notebooks or mobile devices), the peers are considered to be autonomous and heterogeneous [VLO09].

To use a distributed DSMS in such a P2P network with enough performance, reliability, availability and scalability for live sports analysis, multiple mechanisms must be in place. *Query distribution* makes it possible to use more than one peer for query processing (to share system loads). With *fragmentation*, the data streams can be split and processed in parallel on multiple peers. *Replication* increases the reliability of the system by executing a query on multiple peers at the same time. Therefore, results are available, even if a peer failure occurs [Mic14]. *Recovery* also reacts to peer failures in order to restore the previous state of the distributed system at run-time. *Dynamic load balancing* monitors the resource usage of the peers in the network at run-time and shifts continuous queries from one peer to another.

In our opinion, a P2P network of DSMS instances with all features mentioned above can handle the challenges of processing and analyzing data from active sensors in the context of live sport analysis.

3.3 Presentation

The coach wants an agile and lightweight device to access the data. Therefore, we chose mobile devices like smartphones or tables as front-end. These devices are connected to

the P2P network (e.g., via Wi-Fi) and receive the result streams of the continuous queries. The presentation of the calculated sport statistics should be fast and easy to support the coaches' decisions during the game. Therefore, the results have to be aggregated and semantically linked to provide a simple and understandable access to information.

We interviewed different coaches to identify the needed information. We classified those information in (1) player statistics, (2) team statistics and (3) global statistics. *Player statistics* provide information of a specific player (e.g., ball contacts or shots on goal). This view can be used to track the behavior of a single player in significant situations. *Team statistics* provide information about an entire team (e.g., ball possession or pass completion rate). An overview of the current game is provided by the *global statistics* view. All of these views are updated in real-time since the mobile device continuously receives the statistic values.

4 Implementation status

Currently, Herakles is under development and many functions and mechanics are already in place. Therefore, there are no evaluation results available, yet. However, we tested Herakles with the analysis of a self-organized basketball game.

In this section, we give a brief overview of the current state of our implementation. At first, we give a description of our sensor network and data abstraction (section 4.1). The used distributed DSMS and its relevant features are explained in section 4.2. Section 4.3 contains a description about the used sport statistics. Section 4.4 describes the presentation application, which is used for the coach to see the statistics in real-time.

4.1 Sensor networks and data abstraction

We already tested a GPS sensor network for outdoor and a Wi-Fi sensor network for indoor sports. In both sensor networks, position data is provided by mobile applications using sensors of the mobile device. That makes both sensor networks low priced. To test the approach, we equipped 12 players and the ball with a device running the sensor application and let them compete in a short basketball game while recording the resulting data.

The DDC and the different layers of the data abstraction component are implemented. The DDC can be filled in two ways: (1) by reading a file or (2) by messages from other peers. The DDC generates messages to be sent to other DDC instances on other peers resulting in a consistent state through the P2P network. The intermediate schema for the sensor data is shown in listing 1. x, y and z have to be sent by the sensors (but can be converted if they are in a different unit, e.g., in centimeters), whereas ts can be calculated by the DSMS as well as v and a (using two subsequent elements sent by the same sensor). In our opinion, measuring the positions in millimeters and the time in microseconds are sufficient for most sport statistics. Currently, we have implemented adapters to wrap the data streams from the testing source networks mentioned above into our intermediate schema.

Listing 1: Intermediate schema.

```
sid - unique ID
ts  - timestamp [microseconds]
x   - x-position [mm]
y   - y-position [mm]
z   - z-position [mm]
v   - current absolute velocity [mm/s]
a   - current absolute acceleration [mm/s^2]
```

4.2 Distributed Data Stream Management System

For our distributed DSMS, we use *Odysseus*, a highly customizable framework for creating DSMSs. Its architecture consists of easily extensible *bundles*, each of them encapsulating specific functions [AGG+12]. Odysseus provides all basic functions for data stream processing and is already extended for the distributed execution in a P2P network of heterogeneous and autonomous peers. Furthermore, mechanisms for query distribution, fragmentation and replication are already available [Mic14]. Because of its current functionality in addition with the extensibility, we decided to use Odysseus for Herakles.

However, Odysseus had not all features we need to implement a reliable real-time sport analytic system. It did not support dynamic load balancing (for shifting continuous queries at run-time) and recovery. Therefore, we extended Odysseus with these mechanisms.

For the dynamic load balancing, we implemented a communication protocol and a simple load balancing strategy. For recovery, we implemented a combination of active standby and upstream-backup. *Active standby* means to have a continuous query executed multiple times on different peers (similar to replication). However, only the streaming results of one peer are used. But, if that peer fails (or leaves the network on purpose), another peer with a copy of the query replaces it. With *upstream-backup*, a peer saves the stream elements which had been sent to its subsequent peer, until that peer indicates that it has processed said stream elements. If the subsequent peer fails, another peer can install the lost continuous query again and the processing can be redone with the previously saved stream elements.

4.3 Live Statistics

Analyzing the data and calculating useful statistics in terms of continuous queries is an essential part of Herakles. Typically, queries are described by a declarative query language. To avoid complex query declarations on the mobile devices, we implemented an own query language called *SportsQL*. It is a compact language especially designed for sport statistic queries in Herakles. If the user selects a statistic to show, the mobile application of Herakles generates a corresponding SportsQL query, which is sent to an Odysseus-

instance. The transition of the SportsQL query to an executable continuous query is done inside of Odysseus. With this decision, the continuous queries and the mobile application are disconnected from each other: we can create and improve the complex continuous queries in Odysseus without changing the mobile application. Furthermore, any other device can also use SportsQL without changing Odysseus. An example of SportsQL is shown in listing 2.

Listing 2: Example of SportsQL for the shots on goal of a specific player with the id 8.

```
{
    "statisticType": "player",
    "gameType": "soccer",
    "entityId": 8,
    "name": "shotsongoal"
}
```

In this query, a mobile device wants a statistic about the current amount of shots on goal for a specific player. The attribute `name` identifies the statistic to generate and the attribute `gameType` identifies the analyzed game (in this case soccer). `StatisticType` differentiates between team statistic, player statistic and global statistic mentioned in section 3.1. In this example, a player statistic is specified. Therefore, an `entityId` refers to a player for which this statistic should be created. It is possible to send further parameters within a SportsQL query such as time and space parameters. This can be used to limit the query results, e.g., for a specific range of time or for a specific part of the game field.

4.4 Presentation

We implemented an extensible Android application intended for tablets and smartphones. Android has been chosen because it is common and allows the application to be run on different kinds of devices. Figure 3 shows a screenshot of our current application.

5 Future Work

In our current implementation of Herakles, we focus on soccer-specific statistics to show the proof of concept. But there are a lot of possibilities for more advanced statistics and views. Depending on the sport, we want to support additional statistics. Consequently, we plan to enhance our mobile application, the SportsQL query language and the corresponding continuous queries in the P2P network.

Herakles works with radio-based sensors reducing the costs compared to video-based systems. Therefore, it focuses on training sessions or friendly games. Nevertheless, the probability to get a license to use such a system within an official, professional match is low

Figure 3: Mobile application for the coach with statistics and a game topview.

whereas video-based systems are more accepted. An extension could be to use a video-based system to get the player positions.

Currently, we do not consider inaccuracies in sensor data streams. For example, some sensors send inaccurate data or no data at all for a certain time. Therefore, there can be anomalies in sensor data [ACFM14] and sport statistics, which should be considered in Herakles in the future.

Additionally, we do not face the problems of security, which rise with the use of private P2P networks. Currently, we expect that each peer is cooperative and does not want to damage the system on purpose. But this assumption must be weakened in the future: sensor data streams have to be encrypted, peers need to be checked (e.g., web of trust [GS00]) and a distributed user management should be in place.

An open issue is the lack of evaluation. Currently, we are implementing the last steps. We made a few tests with a GPS sensor network and are about to begin with the evaluation. We plan to measure how much data Herakles can process in real-time, how fast it can react to sport-events (like interruptions), how accurate the sport statistics are and how much reliability and availability the P2P network really provides.

6 Summary

In professional sports, complex and expensive computer-based analysis systems are used to collect data, to setup statistics, and to compare players. Most of them are video-based and not every team has the opportunity to buy and maintain them. With Herakles, we proposed an alternative sport analysis system, which uses a decentralized and dynamic P2P network of conventional private computers. A sensor network placed on the playing field (e.g., sensors attached to the players and balls) is continuously sending position data to this P2P network. On each peer, a DSMS is installed. This collection of DSMSs is used to cooperatively process the streaming sensor data in real-time. To be independent

from specific sensors, we designed a data abstraction layer, which separates the sensors from our continuous queries generating the sport statistics. Herakles presents the statistics on a mobile device, where the user immediately sees those statistics during the game. Users can choose other statistics at any time and the P2P network adapts to these changes automatically.

In the P2P network, we use Odysseus, a component-based framework for developing DSMSs. Despite the data stream processing, it supports continuous query distribution, replication and fragmentation. We added dynamic load balancing and recovery mechanisms to fulfill our requirements. Finally, Herakles uses Android-based mobile devices to show the calculated statistics in real-time.

There are still many tasks to do: primarily, extensions to other sports and statistics. We are about to begin with the evaluation to measure performance of efficiency of Herakles. But we are confident: With Herakles, we show that it is possible to analyze sport events in real-time with commodity hardware like notebooks.

References

[ACFM14] Annalisa Appice, Anna Ciampi, Fabio Fumarola und Donato Malerba. *Data Mining Techniques in Sensor Networks - Summarization, Interpolation and Surveillance*. Springer Briefs in Computer Science. Springer, 2014.

[AGG+12] H.-Jürgen Appelrath, Dennis Geesen, Marco Grawunder, Timo Michelsen und Daniela Nicklas. Odysseus: a highly customizable framework for creating efficient event stream management systems. DEBS '12, Seiten 367–368. ACM, 2012.

[Bau00] C.S. Baum. Sports analysis and testing system, 2000. US Patent 6,042,492.

[GS00] T. Grandison und M. Sloman. A survey of trust in internet applications. *Communications Surveys Tutorials, IEEE*, 3(4):2–16, Fourth 2000.

[LMS+08] Marco Leo, Nicola Mosca, Paolo Spagnolo, Pier Luigi Mazzeo, Tiziana D'Orazio und Arcangelo Distante. Real-time multiview analysis of soccer matches for understanding interactions between ball and players. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, Seiten 525–534. ACM, 2008.

[Mic14] Timo Michelsen. Data stream processing in dynamic and decentralized peer-to-peer networks. In *Proceedings of the 2014 SIGMOD PhD symposium*, Seiten 1–5. ACM, 2014.

[SDK+08] Alan F. Smeaton, Dermot Diamond, Philip Kelly, Kieran Moran, King-Tong Lau, Deirdre Morris, Niall Moyna, Noel E O'Connor und Ke Zhang. Aggregating multiple body sensors for analysis in sports. 2008.

[vdGFW+11] Thomas von der Grün, Norbert Franke, Daniel Wolf, Nicolas Witt und Andreas Eidloth. A real-time tracking system for football match and training analysis. In *Microelectronic Systems*, Seiten 199–212. Springer, 2011.

[VLO09] Quang Hieu Vu, Mihai Lupu und Beng Chin Ooi. *Peer-to-Peer Computing: Principles and Applications*. Springer, 2009.

Bestimmung von Datenunsicherheit in einem probabilistischen Datenstrommanagementsystem

Christian Kuka
SCARE-Graduiertenkolleg
Universität Oldenburg
D-26129 Oldenburg
christian.kuka@uni-oldenburg.de

Daniela Nicklas
Universität Bamberg
D-96047 Bamberg
dnicklas@acm.org

Abstract: Für die kontinuierliche Verarbeitung von unsicherheitsbehafteten Daten in einem Datenstrommanagementsystem ist es notwendig das zugrunde liegende stochastische Modell der Daten zu kennen. Zu diesem Zweck existieren mehrere Ansätze, wie etwas das Erwartungswertmaximierungsverfahren oder die Kerndichteschätzung. In dieser Arbeit wird aufgezeigt, wie die genannten Verfahren in ein Datenstrommanagementsystem verwendet werden können, umso eine probabilistische Datenstromverarbeitung zu ermöglichen und wie sich die Bestimmung des stochastischen Modells auf die Latenz der Verarbeitung auswirkt. Zudem wird die Qualität der ermittelten stochastischen Modelle verglichen und aufgezeigt, welches Verfahren unter welchen Bedienungen bei der kontinuierlichen Verarbeitung von unsicherheitsbehafteten Daten am effektivsten ist.

1 Einführung

Für die qualitätssensitive Verarbeitung von Sensordaten ist es notwendig die aktuelle Qualität der Daten zu kennen. Eine der hierbei häufig verwendeten Qualitätsdimensionen ist der statistische Fehler von Sensormessungen. In vielen Fällen wird hierbei die aus dem Datenblatt stammende Kennzahl für die Standardabweichung herangezogen um das stochastische Modell im Sinne einer Normalverteilung zu verwenden. Jedoch kann das Rauschen eines Sensors von vielen Kriterien abhängen und sich vor allem auch dynamisch zur Laufzeit ändern. Eine Form der Qualitätsbestimmung besteht darin, direkt das zugrunde liegende stochastische Modell der Sensormessungen kontinuierlich neu zu ermitteln. Vor allem im Bereich der kontinuierlichen Verarbeitung von hochfrequenten Sensordaten ist es hierbei notwendig die Speicherkapazitäten des Systems zu beachten und die Daten so schnell wie möglich zu verarbeiten. Für diese Form der Verarbeitung existiert mittlerweile eine Vielzahl von Systemen, welche unter dem Begriff Datenstrommanagementsystem zusammengefasst werden können.

Im Rahmen von Datenstrommanagementsystemen hat sich für die Verarbeitung von Unsicherheiten der Begriff der probabilistischen Datenstromverarbeitung [TPD+12, JM07, KD09] etabliert. Ziel der Verarbeitung ist es nicht nur den reinen Messwert, sondern die zugrunde liegende Unsicherheit innerhalb der Verarbeitung in einem Datenstrommanagementsystem zu repräsentieren und zu verarbeiten, so dass der entstehende kontinuierliche

Ausgabestrom einer Anfrage auch immer die aktuelle Ergebnisunsicherheit enthält. Bei der Verarbeitung von Unsicherheiten kann dabei zwischen zwei Klassen unterschieden werden, der Verarbeitung von diskreten Wahrscheinlichkeitsverteilungen und der Verarbeitung von kontinuierlichen Wahrscheinlichkeitsverteilungen. Diskrete Verteilungen werden häufig dazu genutzt die Existenzunsicherheit von möglichen Welten darzustellen. Kontinuierliche Wahrscheinlichkeitsverteilungen dienen dagegen dazu, Unsicherheiten in der Sensorwahrnehmung, welche etwa durch das Messverfahren an sich oder Umwelteinflüsse induziert werden, zu beschreiben. Im Folgenden liegt der Fokus daher auf der Bestimmung von kontinuierlichen stochastischen Modellen.

Die Bestimmung von stochastischen Modellen auf Basis von Datenströmen bei Filteroperationen wurde unter anderem in [ZCWQ03] behandelt. Hierbei war allerdings das Ziel, das stochastische Modell zu verwenden, um das Rauschen um einen Selektionsbereich innerhalb der Verarbeitung zu bestimmen. Ziel dieser Arbeit ist es aber das mehrdimensionale stochastische Modell der Daten selbst zu bestimmen, um eine probabilistische Verarbeitung der Daten, wie sie in [TPD+12] mit dem Mischtyp-Modell eingeführt wurde, zu ermöglichen. Das Modell hat den Vorteil, dass es sowohl die Unsicherheit über die Existenz einzelner Attribute, sowie auch die Unsicherheit über die Existenz ganzer Tupel repräsentieren kann. Zur Evaluation von verschiedenen Verfahren zur Bestimmung und Verarbeitung der mehrdimensionalen stochastischen Modelle wurde diese probabilistische Verarbeitung mit den Konzepten der deterministischen Verarbeitung mit Zeitintervallen aus [Krä07] kombiniert und in dem Datenstrommanagementsystem Odysseus [AGG+12] implementiert.

2 Verfahren zur Bestimmung von stochastischen Modellen

Für die Bestimmung von mehrdimensionalen stochastischen Modellen, wie sie bei der probabilitischen Datenstromverarbeitung verwendet werden, existieren prinzipiell mehrere Möglichkeiten. Zu diesen Verfahren zählen etwa das Erwartungswertmaximierungsverfahren und die Kerndichteschätzung, welche im Folgenden näher erläutert werden.

2.1 Erwartungsmaximierungsverfahren

Das Erwartungswertmaximierungsverfahren [DLR77] dient dazu die Parameter eines stochastischen Modells durch mehrere Iterationen an die Verteilung von Daten anzunähern. Hierzu wird versucht die Log-Likelihood L zwischen den zu bestimmenden Parametern und den zur Verfügung stehenden Daten in jeder Iteration t des Algorithmus zu maximieren. Als Parameter bieten sich hierfür die Parameter einer multivariaten Mischverteilung aus Gauß-Verteilungen mit Parameter $\theta = \{w_i, \mu_i, \Sigma_i\}_{i=1}^{m}$ an. Eine multivariate Mischverteilung aus Gauß-Verteilungen über eine kontinuierliche Zufallsvariable X ist eine Menge

von m gewichteten Gauß-Verteilungen $X_1, X_2, \ldots, X_m$, wobei X_i die Wahrscheinlichkeitsdichtefunktion

$$f_X(x) = \sum_{i=1}^{m} w_i f_{X_i}(x) \text{ mit } f_{X_i}(x) = \frac{1}{(2\pi)^{k/2} \mid \Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}$$

ist. Dabei gilt, dass $0 \leq w_i \leq 1, \sum_{i=1}^{m} w_i = 1$, k die Größe des Zufallsvektors ist und jede Mischverteilungskomponente X_i eine k-variate Gauß-Verteilung mit Erwartungswert μ_i und Kovarianz-Matrix Σ_i ist. Zur Annäherung einer Gauß-Mischverteilung wird zunächst ein initiales stochastisches Modell mit m Gauß-Verteilungen bestimmt. Auf Basis des aktuellen Modells werden nun im E-Schritt die Erwartungswerte bestimmt, also die Wahrscheinlichkeiten, dass die aktuellen Werte aus dem aktuellen stochastischen Modell generiert wurden.

$$\tau_{ij}^{(t)} = \frac{w_j^{(t)} F_{Xj}(x_i; \theta_j^{(t)})}{\sum_{l=1}^{m} w_l^{(t)} F_{Xj}(x_i; \theta_l^{(t)})}, i = 1, \ldots, n, j = 1, \ldots, m$$

$$\gamma_j^{(t)} = \sum_{i=1}^{n} \tau_{ij}^{(t)}, j = 1, \ldots, m$$

Während des M-Schrittes werden die neuen Parameter für θ anhand der Ergebnisse aus dem E-Schritt bestimmt.

$$w_j^{(t+1)} = \frac{\gamma_j^{(t)}}{n}, j = 1, \ldots, m, \ \mu_j^{(t+1)} = \frac{1}{\gamma_j^{(t)}} \sum_{i=1}^{n} \tau_{ij}^{(t)}, j = 1, \ldots, m$$

$$\Sigma_j^{(t+1)} = \frac{1}{\gamma_j^{(t)}} \sum_{i=1}^{n} \tau_{ij}^{(t)} (x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^T, j = 1, \ldots, m$$

Nach jedem EM-Schritt wird die Log-Likelihood berechnet und mit einem gegebenen Schwellwert verglichen. Ist die Differenz kleiner als der gegebene Schwellwert oder überschreitet die Anzahl der Iterationen die maximale Anzahl, werden die bestimmten Parameter für die Gewichte (w), den Erwartungswert (μ), sowie die Kovarianz-Matrix (Σ) der Mischverteilung zurückgeliefert.

2.2 Kerndichteschätzung

Im Gegensatz zum EM-Verfahren wird bei der Kerndichteschätzung (KDE) für jeden Messwert eine Komponente in einer Mischverteilung erstellt und eine Bandbreite bestimmt. Die Bandbreite dient dazu eine Varianz/Kovarianz Matrix für alle Komponenten der Mischverteilung zu bilden und so das eigentliche zugrunde liegende Modell möglichst gut wieder zu geben. Zur Bestimmung der Bandbreite B haben sich mehrere Verfahren

etabliert, wie etwa die Scott-Regel [Sco92]. Die Parameter der Komponenten der Mischverteilung lassen sich somit wie folgt berechnen:

$$w_j = \frac{1}{n}, \mu_j = x_j, \Sigma_j = \Sigma(x) * B$$

Wobei $\Sigma(x)$ die Varianz/Kovarianz der zugrunde liegenden Daten repräsentiert. Man sieht bereits, dass das KDE ohne mehrmalige Iterationen über die zugrunde liegenden Daten auskommt, da sowohl der Erwartungswert wie auch die Varianz/Kovarianz inkrementell bestimmt werden können.

Da bei der Kerndichteschätzung die Anzahl an Komponenten der Mischverteilung linear zu der Zahl der Messwerte steigt und somit das Ergebnis generell ungeeignet ist für eine Verarbeitung in einem Datenstrommanagementsystem, benötigt es ein Verfahren zur Reduktion der Komponenten. In [ZCWQ03] stellen die Autoren ein Verfahren vor, welches das KDE-Verfahren auf einen eindimensionalen Strom anwendet und die dabei resultierende Mischverteilung durch ein Kompressionsverfahren auf eine geringere Anzahl von Verteilungen reduziert. Dieses Verfahren ist allerdings nicht für multivariate Verteilungen anwendbar. In [CHM12] wurde eine Selbstorganisierende Merkmalskarten (SOM) verwendet um Cluster zu bilden und diese Cluster durch eine Verteilung darzustellen. SOMs haben allerdings allgemein den Nachteil, dass die Gefahr einer Überanpassung der Gewichtsvektoren besteht. Eine weitere Möglichkeit zur Reduktion der Komponenten besteht in dem Bregman Hard Clustering Verfahren [BMDG05]. Bei dem Bregman Hard Clustering Verfahren wird versucht, ähnliche Verteilungen innerhalb einer Mischverteilung durch die Bildung von Clustern zu vereinfachen. Hierbei werden zunächst Cluster mit je einem Repräsentanten gebildet und anschließend für jedes Cluster eine Minimierung ausgeführt mit dem Ziel den Informationsverlust zwischen den Clusterzentren und den Komponenten zu minimieren. Das Verfahren kann als eine Generalisierung des Euklidischen k-Means Verfahrens angesehen werden, wobei die Kullback-Leibler Divergenz als Minimierungsziel verwendet wird. Um allerdings die Bestimmung des Integrals innerhalb der Kullback-Leibler Divergenz zu umgehen, wird die Kullback-Leibler Divergenz in eine Bregman Divergenz umgewandelt. Die Bregman Divergenz ist dabei definiert als:

$$D_F(\theta_j||\theta_i) = F(\theta_j) - F(\theta_i) - < \theta_j - \theta_i, \nabla F(\theta_i) > \tag{1}$$

Hierbei wird die Dichtefunktion einer Normalverteilung in die kanonische Dekomposition der jeweiligen Exponentialfamilie wie folgt umgeschrieben:

$$\mathcal{N}(x; \mu, \sigma^2) = exp\{< \theta, t(x) > -F(\theta) + C(x)\} \tag{2}$$

Wobei $\theta = (\theta_1 = \frac{\mu}{\sigma^2}, \theta_2 = -\frac{1}{2\sigma^2})$ die natürlichen Parameter, $t(x) = (x, x^2)$ die notwendige Statistik und $F(\theta) = -\frac{\theta_1^2}{4\theta_2} + \frac{1}{2}\log\frac{-\pi}{\theta_2}$ die Log-Normalisierung für eine Normalverteilung darstellen.

Unter der Bedingung, dass beide Verteilungen von der gleichen Exponentialfamilie stammen, lässt sich die Kullback-Leibler Divergenz in die Bregman Divergenz umformen

$$KL(\mathcal{N}(x; \mu_i, \sigma_i^2)||\mathcal{N}(x; \mu_j, \sigma_j^2)) = D_F(\theta_j||\theta_i) \tag{3}$$

, so dass nun direkt die Bregman Divergenz als Distanz innerhalb des k-Means Verfahrens zur Clusterbildung angewendet werden kann.

3 Evaluation der Verfahren

Im Folgenden werden die Verfahren zur Bestimmung des stochastischen Modells der Daten eines Datenstrom hinsichtlich ihrer Latenz, aber auch hinsichtlich der Güte des stochastischen Modells evaluiert. Zu diesem Zweck wurden die Verfahren als Verarbeitungsoperatoren innerhalb des Odysseus DSMS realisiert. Die Evaluation wurde dabei sowohl auf synthetischen Daten, wie auch auf Daten aus einem Ultrabreitband-Positionierungssystem [WJKvC12] durchgeführt. Hierzu wurden 10.000 Messwerte aus einer Normalverteilung, sowie aus einer logarithmischen Normalverteilung generiert um einen Datenstrom aus Messwerten zu simulieren. Die Evaluation der Latenz und der Güte des Modells betrachtet dabei drei Szenarien mit Datenfenstern der Größe 10, 100 und 1000. Das Datenfenster definiert dabei die Anzahl an Messwerten auf denen die Operatoren das stochastische Modell bestimmen sollen. Die Güte des Modells betrachtet das aktuell bestimmte stochastische Modell im Hinblick auf alle 10.000 Datensätze. Als Qualitätskriterium wird hierzu das Akaike-Informationskriterium verwendet. Das AIC ist ein Maß für die relative Qualität eines stochastischen Modells für eine gegebene Datenmenge und ist definiert als:

$$AIC = 2k - 2\ln(L) \tag{4}$$

Der Parameter k repräsentiert hierbei die Anzahl der freien Parameter in dem stochastischen Modell und der Parameter L gibt die Log-Likelihood zwischen dem stochastischen Modell und der gegebenen Datenmenge wieder. Dieses Informationskriterium ist für die Evaluation deshalb gut geeignet, da es sowohl die Nähe der generierten Mischverteilung aus den drei Verfahren zu den tatsächlichen Daten bewertet und zudem die Anzahl der Komponenten innerhalb der Mischverteilungen in die Bewertung mit einfließen lässt. Die Nähe zu den tatsächlichen Daten ist wichtig für die Qualität der Verarbeitungsergebnisse und die Komponentenanzahl der Mischverteilung hat eine Auswirkung auf die Latenz der Verarbeitung, da jede Komponente innerhalb einer Mischverteilung bei Operationen wie der Selektion oder dem Verbund mit einem Selektionskriterium bei einer probabilistischen Verarbeitung integriert werden muss.

Um mögliche Ausreißer zu minimieren wurde jede Evaluation wurde dabei 10-mal wiederholt. Als Testsystem diente ein Lenovo Thinkpad X240 mit Intel Core i7 und 8GB RAM. Die verwendete Java Laufzeitumgebung war ein OpenJDK Runtime Environment (IcedTea 2.5.2) (7u65-2.5.2-2) mit einer OpenJDK 64-Bit Server VM (build 24.65-b04, mixed mode). Bei dem Betriebssystem handelte es sich um ein Debian GNU/Linux mit einem 3.14 Kernel.

Das EM-Verfahren versucht ein stochastisches Modell an die eingehenden Daten anzupassen. Dabei spielen neben der Datenfenstergröße die Anzahl der Iterationen, der Konvergenzschwellwert für die Veränderung der Log-Likelihood in jeder Iteration, sowie die Komponentenanzahl der Mischverteilungen eine Rolle für die Latenz dieses Operators. Für die Evaluation wurde der Konvergenzschwellwert auf 0.001 gesetzt, die Anzahl an Iterationen auf 30 und die Zahl der Komponenten auf 2. Die gleiche Anzahl an Iterationen wird ebenfalls in der von V. Garcia bereitgestellten Java Bibliothek jMEF[1] verwendet.

[1] http://vincentfpgarcia.github.io/jMEF/

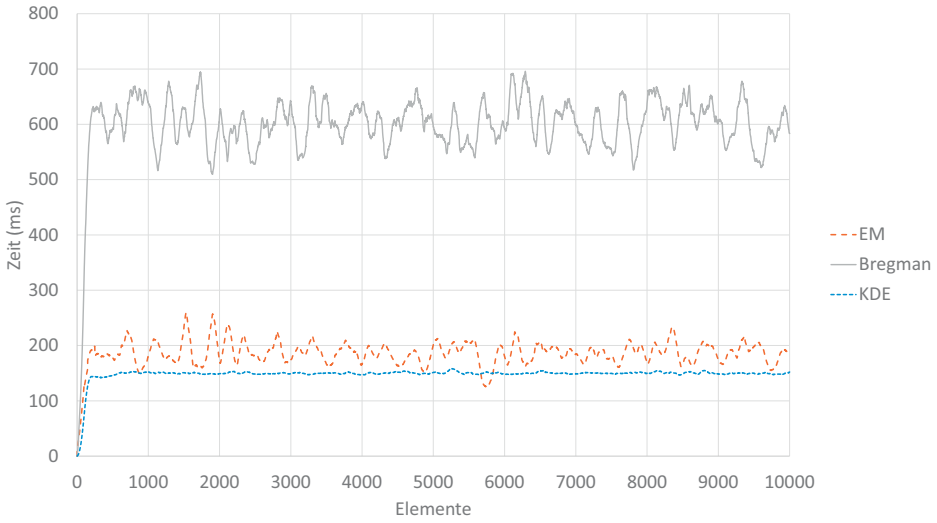


Abbildung 1: Latenz der Operatoren bei einem Datenfenster der Größe 100 für Daten aus einer logarithmischen Normalverteilung

Das KDE-Verfahren bestimmt für jeden Datenwert eine eigene Komponente in der resultierenden Mischverteilung. Der entwickelte Operator verwendet hierzu die Scott-Regel zur Bestimmung der Bandbreite der Kovarianzmatrix der Komponenten. Das Bregman Hard Clustering, welches in einem weiteren Schritt verwendet wird um die Anzahl an Komponenten auf die gewünschte Zahl zu minimieren wurde mit einer maximalen Anzahl von 30 Iterationen konfiguriert. Um die Resultate vergleichbar zu halten wurde der Operator so konfiguriert, dass er ebenfalls eine 2-komponentige Mischverteilung ermittelt, also zwei Cluster bildet.

Der hier verwendete Konvergenzschwellwert für das Erwartungswertmaximierungsverfahren liegt oberhalb des, in der verwendeten Apache Commons Math3 Bibliothek[2] als Standardwert festgelegten, Wertes von 0.00001, da sich in den Versuchen zeigte, dass bereits ein höherer Konvergenzschwellwert ausreichte um die Verfahren hinsichtlich der Güte des stochastischen Modells und der gemessenen Latenz miteinander zu vergleichen.

3.1 Synthetische Sensordaten

Das Latenzverhalten der einzelnen Verfahren ist in Abb. 1 für Daten aus einer log. Normalverteilung für ein Datenfenster der Größe 100 dargestellt. Das EM-Verfahren weist hierbei eine ähnliche und stabile Latenz von durchschnittlich ca. 200 Millisekunden auf. Dies ist durch die mehrmalige Iteration über die aktuell gültigen Daten zur Bestimmung der

[2] http://commons.apache.org/proper/commons-math

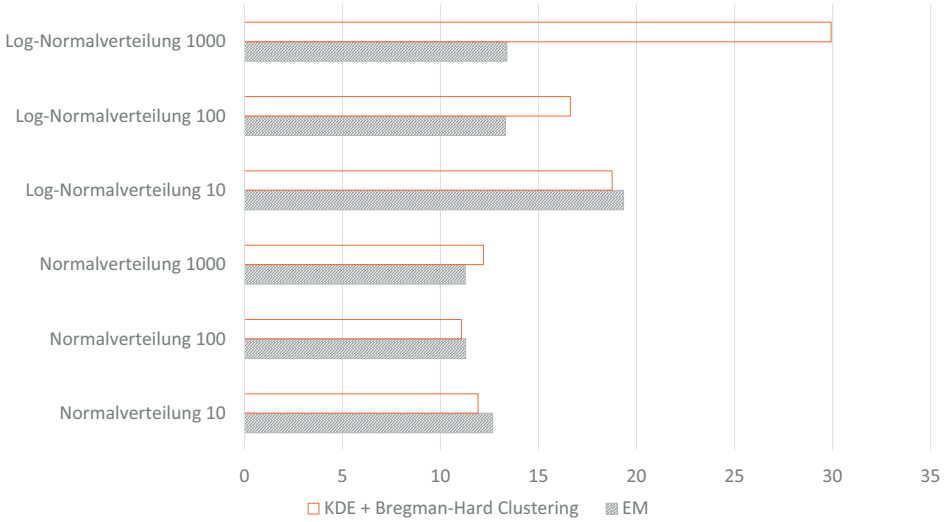


Abbildung 2: Vergleich des AIC zwischen EM-Verfahren und KDE mit Bregman Hard Clustering bei unterschiedlichen Datensatzfenstergrößen für Werte aus einer Normalverteilung und einer logarithmischen Normalverteilung

Log-Likelihood zwischen dem jeweils temporären stochastischen Modell und den Daten geschuldet. Im Gegensatz zum EM-Verfahren kann das Band bei der Kerndichteschätzung kontinuierlich bestimmt werden. Allerdings fällt auf, dass trotz mehrmaliger Wiederholung der Messung das Verfahren zum Bregman Hard Clustering eine deutlich höhere Latenz aufweist. Dieses Verhalten ist dabei unabhängig von der Art der Verteilung. Dies ist vor allem auf die Tatsache zurück zu führen, dass das Bregman Hard Clustering Verfahren in jeder Iteration die Bregman Divergenz zwischen den Clusterzentren und den einzelnen Komponenten bestimmen muss und zusätzlich noch den Zentroiden aus jedem Cluster in jeder Iteration neu ermitteln muss. Beim Vergleich der durchschnittlichen Latenz bei unterschiedlichen Größen von Datenfenstern zeigt sich, dass die Latenz des EM-Verfahrens konstant bleibt, während die Latenz des Bregman Hard Clusterings stark ansteigt. Bei der Qualitätsbetrachtung des ermittelten stochastischen Modells fällt auf, dass die Qualität des EM-Verfahrens im Sinne des AIC bei Werten aus einer logarithmischen Normalverteilung deutlich besser abschneidet als das KDE-Verfahren in Kombination mit dem Bregman Hard Clustering. Bei Werten aus einer Normalverteilung dagegen unterscheidet sich der AIC-Wert nur geringfügig bei den beiden Verfahren. Ein gleiches Verhalten lässt sich auch bei Datensatzfenstern der Größe 1.000 beobachten.

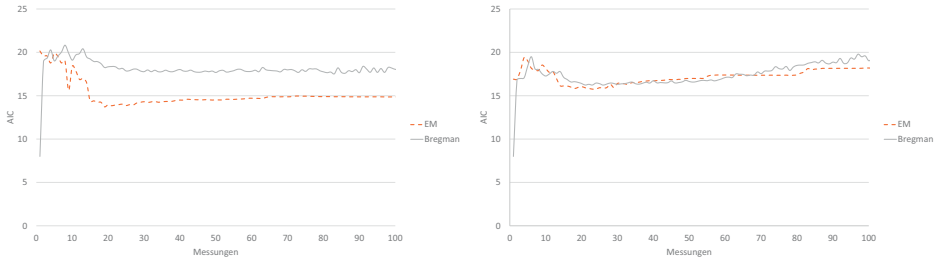
Ist allerdings die Anzahl an Datensätzen gering, ändert sich dieses Verhalten. Bei einem Datensatzfenster der Größe 10 zeigt sich unabhängig von dem zugrunde liegenden stochastischen Modell der Daten, dass die Kombination aus KDE und Bregman Hard Clustering das bessere Modell liefert. Zudem unterscheiden sich die Latenzen bei dieser Datenmenge zwischen den beiden Verfahren nur gering.

Abbildung 3: Messwerte der Positionsbestimmung für die Positionen 1–8

3.2 Reale Sensordaten

Um zu zeigen, dass die Verfahren auch stochastische Modelle von echten Sensordaten erstellen können, wurden die Operatoren auf Sensordatenaufzeichnungen eines Ultrabreitband-Positionierungssystem [WJKvC12] angewendet. Insgesamt wurden 8 Positionen (vgl. Abbildung 3) bestimmt, von denen im Folgenden die Positionen 6 und 7 als repräsentative Positionen näher betrachtet werden. Hierbei wurde das stochastische Modell jeder Position mit dem EM-Verfahren und der Kombination aus KDE und Bregman Hard Clustering auf einem Datensatzfenster der Größe 10 und einem Datensatzfenster der Größe 100 bestimmt.

Bei der Betrachtung der zeitlichen Bestimmung des stochastischen Modells in Abb. 4 fallen zunächst für die Position 6 anfängliche Ausreißer bei der Nähe zum Modell auf. Dies deutet auf eine anfängliche Anpassung der Positionierungsknoten der Anwendung hin. In den darauf folgenden Messungen bleiben sowohl die Modellqualität des EM-Verfahrens, wie auch das resultierende Modell des Bregman Hard Clustering stabil. Wie bereits bei den synthetischen Daten ist auch bei realen Sensordaten das Phänomen erkennbar, dass die Kombination aus KDE mit Bregman Hard Clustering bei kleinen Datensatzfenstern im Vergleich zum EM-Verfahren bessere stochastische Modelle ermittelt. Dagegen ist bei größeren Datensatzfenstern das EM-Verfahren besser geeignet um gute stochastische Modell im Sinne des AIC zu bestimmen.



(a) Position 6

(b) Position 7

Abbildung 4: Qualität des stochastischen Modells über die Zeit bei einem Datensatzfenster der Größe 100 von Position 6 und 7

4 Zusammenfassung und Ausblick

In dieser Arbeit wurden Verfahren zur kontinuierlichen Bestimmung des zugrunde liegenden mehrdimensionalen stochastischen Modells von Messwerten aus aktiven Datenquellen vorgestellt. Ziel ist es, diese mehrdimensionalen stochastischen Modelle in einem probabilistischen Datenstrommanagementsystem zu verarbeiten. Bei den Verfahren handelt es sich um das Erwartungsmaximierungsverfahren und die Kerndichteschätzung in Kombinationen mit dem Bregman Hard Clustering Ansatz. Zunächst wurden die Grundlagen der jeweiligen Verfahren aufgezeigt. Zur Repräsentation der Unsicherheiten wurde das in [Krä07] entwickelte Modell durch das Mischtyp Modell [TPD+12] erweitert und in dem Odysseus DSMS realisiert. Bei der Evaluation der Verfahren wurde zunächst auf Basis von synthetischen Daten die Latenz der einzelnen Verfahren ermittelt. Hierbei zeigte sich, dass die Kombination aus Kerndichteschätzung und Bregman Hard Clustering aufgrund der mehrmaligen Iterationen über die Komponenten einer Mischverteilung eine wesentlich höhere Latenz als das Erwartungsmaximierungsverfahren aufweist. Zudem sind die resultierenden stochastischen Modelle im Sinne des Akaike Informationskriterium in den meisten Fällen schlechter als die angenäherten Modelle des Erwartungsmaximierungsverfahrens. Aus Sicht der Latenzoptimierung und angesichts der Qualität der bestimmten Modelle sollte daher das Erwartungsmaximierungsverfahren bei der Datenstromverarbeitung bevorzugen werden. Einzige Ausnahme sind Anwendungen in denen nur geringe Mengen an Daten zur Verfügung stehen. Hier konnte die Kombination aus Kerndichteschätzung und Bregman Hard Clustering die besseren stochastischen Modelle bestimmen. Eine Evaluation auf Basis von Sensoraufzeichnungen von Ultrabreitband-Lokalisierungssensoren bestätigten die Resultate aus der Evaluation mit synthetischen Daten.

Danksagung

Die Autoren möchten Herrn Prof. Huibiao Zhu von der East China Normal University für seine Unterstützung danken. Diese Arbeit wurde durch die Deutsche Forschungsgesellschaft im Rahmen des Graduiertenkollegs (DFG GRK 1765) SCARE (www.scare.uni-oldenburg.de) gefördert.

Literatur

[AGG+12] H.-J. Appelrath, Dennis Geesen, Marco Grawunder, Timo Michelsen und Daniela Nicklas. Odysseus: a highly customizable framework for creating efficient event stream management systems. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, Seiten 367–368, New York, NY, USA, 2012. ACM Press.

[BMDG05] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon und Joydeep Ghosh. Clustering with Bregman Divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[CHM12] Yuan Cao, Haibo He und Hong Man. SOMKE: Kernel density estimation over data streams by sequences of self-organizing maps. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1254–1268, 2012.

[DLR77] Arthur P. Dempster, Nan M. Laird und Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.

[JM07] T. S. Jayram und S. Muthukrishan. Estimating statistical aggregates on probabilistic data streams. In *In ACM Symposium on Principles of Database Systems*, Seiten 243–252, New York, NY, USA, 2007. ACM Press.

[KD09] Bhargav Kanagal und Amol Deshpande. Efficient query evaluation over temporally correlated probabilistic streams. In *International Conference on Data Engineering*, 2009.

[Krä07] Jürgen Krämer. *Continuous Queries over Data Streams-Semantics and Implementation*. Dissertation, Philipps-Universität Marburg, 2007.

[Sco92] D.W Scott. Multivariate Density Estimation: Theory, Practice, and Visualization. 1992.

[TPD+12] Thanh T. L. Tran, Liping Peng, Yanlei Diao, Andrew McGregor und Anna Liu. CLARO: modeling and processing uncertain data streams. *The VLDB Journal*, 21(5):651–676, Oktober 2012.

[WJKvC12] Thorsten Wehs, Manuel Janssen, Carsten Koch und Gerd von Cölln. System architecture for data communication and localization under harsh environmental conditions in maritime automation. In *Proceedings of the 10th IEEE International Conference on Industrial Informatics (INDIN)*, Seiten 1252–1257, Los Alamitos, CA, USA, 2012. IEEE Computer Society.

[ZCWQ03] Aoying Zhou, Zhiyuan Cai, Li Wei und Weining Qian. M-kernel merging: Towards density estimation over data streams. In *8th International Conference on Database Systems for Advanced Applications*, Seiten 285–292. IEEE, 2003.

Kontinuierliche Evaluation von kollaborativen Recommender-Systeme in Datenstrommanagementsystemen
– Extended Abstract –

Cornelius A. Ludmann, Marco Grawunder, Timo Michelsen, H.-Jürgen Appelrath

University of Oldenburg, Department of Computer Science
Escherweg 2, 26121 Oldenburg, Germany
{cornelius.ludmann, marco.grawunder, timo.michelsen, appelrath}@uni-oldenburg.de

Recommender-Systeme (RecSys) findet man in vielen Informationssystemen. Das Ziel eines RecSys ist das Interesse eines Benutzers an bestimmten Objekten (engl. *item*) vorherzusagen, um aus einer großen Menge an Objekten diejenigen dem Benutzer zu empfehlen, für die das vorhergesagte Interesse des Benutzers am größten ist. Die zu empfehlenden Objekte können zum Beispiel Produkte, Filme/Videos, Musikstücke, Dokumente, Point of Interests etc. sein. Das Interesse eines Benutzers an einem Objekt wird durch eine *Bewertung* (engl. *rating*) quantifiziert. Die Bewertung kann explizit durch den Benutzer angegeben (der Benutzer wird dazu aufgefordert, ein bestimmtes Objekt zu bewerten) oder implizit vom Verhalten des Benutzers abgeleitet werden (im einfachsten Fall durch eine binäre Bewertung: Objekt genutzt vs. nicht genutzt). Mit Methoden des maschinellen Lernens wird aus bekannten Bewertungen ein Modell trainiert, dass unbekannte Bewertungen vorhersagen kann. Für die Bestimmung der Empfehlungsmenge werden die Bewertungen aller unbewerteten Objekte für einen Benutzer vorhergesagt und die best-bewerteten Objekte empfohlen.

Im realen Einsatz eines RecSys entstehen kontinuierlich neue Bewertungen, die bei der Integration in das Modell die Vorhersagen für alle Benutzer verbessern können. Betrachtet man die Bewertungs- und Kontextdaten nicht als statische Lerndaten, sondern als kontinuierlich auftretende, zeitannotierte Events eines potenziell unendlichen Datenstroms, entspricht das eher der Situation, wie ein RecSys produktiv eingesetzt wird. Zur Umsetzung eines RecSys schlagen wir die Erweiterung eines Datenstrommanagementsystems (DSMS) vor, welches mit Hilfe von Datenstrom-Operatoren und kontinuierlichen Abfrageplänen Datenströme mit Lerndaten zum kontinuierlichen Lernen eines Modells nutzt. Die Anwendung des Modells zur Bestimmung der Empfehlungsmenge wird ebenso durch Events eines Datenstroms ausgelöst (zum Beispiel durch die Anzeige in einer Benutzerapplikation).

Integriert man ein RecSys in ein DSMS, so stellt sich die Frage, wie verschiedene Ansätze bzw. Implementierungen evaluiert und verglichen werden können. Bei der Evaluation werden die Daten i. d. R. in Lern- und Testdaten aufgeteilt. Die Evaluation von RecSys, bei denen der zeitliche Zusammenhang der Daten eine Rolle spielt, hat besondere Anfor-

derungen: Die zeitliche Reihenfolge der Lerndaten muss erhalten bleiben und es dürfen keine Testdaten zur Evaluation genutzt werden, die zeitlich vor den genutzten Lerndaten liegen. Um den zeitlichen Verlauf zu berücksichtigen, wird ein Datenstrommodel genutzt, welches die Nutzdaten mit Zeitstempeln annotiert.

Zur Evaluierung eines DSMS-basierten RecSys schlagen wir den in Abbildung 1 dargestellten Aufbau vor. Als Eingabe erhält das DSMS als Bewertungsdaten die Tupel $(u, i, r)_t$ mit der Bewertung r des Benutzers u für das Objekt i zum Zeitpunkt t sowie Anfragen für Empfehlungen für den Benutzer u zum Zeitpunkt t. Der Evaluationsaufbau gliedert sich grob in drei Teile: *Continuous Learning* nutzt die Lerndaten zum maschinellen Lernen des RecSys-Modells. *Continuous Recommending* wendet zu jeder Empfehlungsanfrage das Modell an, um für den entsprechenden Benutzer eine Empfehlungsmenge auszugeben. *Continuous Evaluation* teilt die Bewertungsdaten in Lern- und Testdaten auf und nutzt das gelernte Modell zur Evaluation. Dazu wird die Bewertung für ein Testtupel vorhergesagt und die Vorhersage mit der wahren Bewertung verglichen.

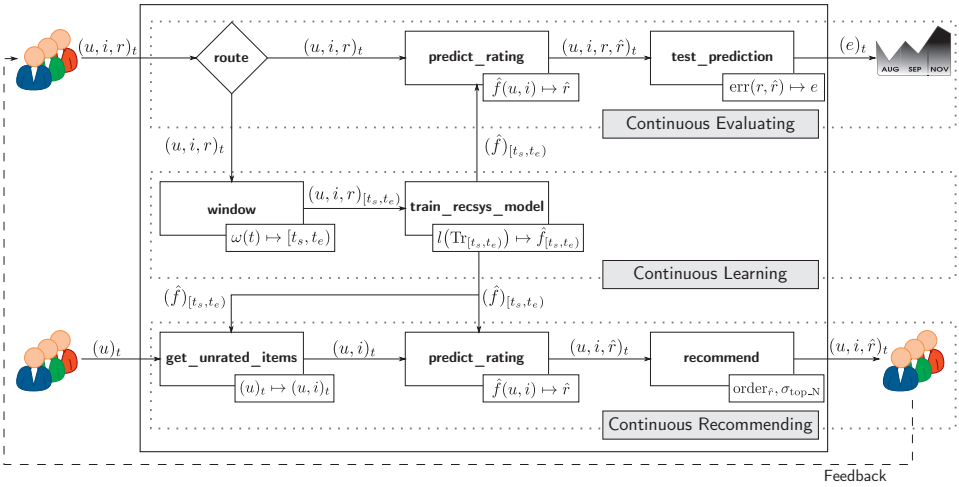


Abbildung 1: Aufbau eines RecSys mit kontinuierlicher Evaluation

Dieser Aufbau wurde mit dem DSMS Odysseus [AGG+12] prototypisch umgesetzt und die Evaluation mit dem MovieLens-Datensatz[1] durchgeführt. Als nächste Schritte sollen weitere Evaluationsmethoden mit diesem Aufbau umgesetzt, sowie Algorithmen zum maschinellen Lernen für RecSys für den Einsatz in einem DSMS optimiert werden.

Literatur

[AGG+12] H.-Jürgen Appelrath, Dennis Geesen, Marco Grawunder, Timo Michelsen und Daniela Nicklas. Odysseus: A Highly Customizable Framework for Creating Efficient Event Stream Management Systems. In *DEBS'12*, Seiten 367–368. ACM, 2012.

[1] `http://grouplens.org/datasets/movielens/`

Using Data-Stream and Complex-Event Processing to Identify Activities of Bats

Extended Abstract

Sebastian Herbst, Johannes Tenschert, Klaus Meyer-Wegener
Data Management Group, FAU Erlangen-Nürnberg, Erlangen, Germany
E-Mail <firstname>.<lastname>@fau.de

1 Background and Motivation

Traditional tracking of bats uses telemetry [ADMW09]. This is very laborious for the biologists: At least two of them must run through the forest to get a good triangulation. And, only one bat at a time can be tracked with this method.

The Collaborative Research Center 1508 of the German Science Foundation (DFG) has been established to develop more sophisticated sensor nodes for the bats to carry[1]. These *mobile nodes* must not be heavier than the telemetry senders used so far, but they offer much more processing capacity in addition to sending a beacon with an ID.

Ground nodes receive the signals transmitted by the mobile nodes. They are also sensor nodes that run on batteries. Currently, all their detections are forwarded to a central *base station*, where a localization method integrates them into a position estimation for each bat [NKD+15]. The base station is a standard computer with sufficient power and energy. In the future, some parts of the localization may already be done on the ground nodes to reduce the data transmission and thus save some energy.

Output of the localization method is a *position stream*. Each element of this stream contains a timestamp, a bat ID, and x and y coordinates. While the biologists would like to have a z coordinate as well, it cannot be provided at the moment because of technological restrictions. Plans are to include it in the future. While filtering and smoothing have already been done, the precision of the coordinate values depends on the localization method used. It may only be in the order of meters to tens of meters, or (with much more effort) in the range of decimeters. Furthermore, some positions may be missing in the stream. A bat may be temporarily out of the range of the ground nodes, or its mobile node may be switched off to save energy. Hence, the subsequent processing must be robust with respect to imprecise position values.

[1] The sensor nodes are glued at the neck of the bats. After two to four weeks, they fall off.

2 Goals and Challenges

The goal of this work is to investigate the use of of data-stream processing (DSP) and complex-event processing (CEP) to extract information meaningful for biologists from the position stream described. Biologists are interested in patterns of bat behavior, which are known to some extent, but have not been observed over longer periods of time, and have not been correlated in time for a group of bats. The elementary parts of the patterns can be expressed in terms of flight trajectories, which are a special case of semantic trajectories [PSR+13]. Biologists however are more interested in bat activities indicated by sequences of trajectories.

Our idea is to identify these activities with the near-real-time processing that a combination of DSP and CEP can provide. This gives information on current activities earlier to the biologists, so they could go out and check by themselves what is happening. Also, they can experiment with the bats by providing extra food or emitting sounds. Furthermore, the mobile nodes on the bats can be configured to some extent, as can be the ground nodes. So when a particular behavior is reported to the biologists, they can adjust the localization method, e. g. switch to higher precision, even if that costs more energy.

In order to reach that goal, a first set of DSP queries and CEP rules has been defined. Activities are managed as objects for each bat. A change of activity is triggered by events and is recorded as an update of this object. Current activities as well as the previous activities of each bat are displayed for the biologists. Ongoing work evaluates the activity detection by comparing the output with the known activities of bats in a simulation tool. It has already helped to adjust the substantial number of parameter values used in the DSP queries and CEP rules.

Acknowledgments. This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) under the grant of FOR 1508 for sub-project no. 3.

References

[ADMW09] Sybill K. Amelon, David C. Dalton, Joshua J. Millspaugh, and Sandy A. Wolf. Radiotelemetry; techniques and analysis. In Thomas H. Kunz and Stuart Parsons, editors, *Ecological and behavioral methods for the study of bats*, pages 57–77. Johns Hopkins University Press, Baltimore, 2009.

[NKD+15] Thorsten Nowak, Alexander Koelpin, Falko Dressler, Markus Hartmann, Lucila Patino, and Joern Thielecke. Combined Localization and Data Transmission in Energy-Constrained Wireless Sensor Networks. In *Wireless Sensors and Sensor Networks (WiSNet), 2015 IEEE Topical Conference on*, Jan 2015.

[PSR+13] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady Andrienko, Natalia Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, Jose Macedo, Nikos Pelekis, Yannis Theodoridis, and Zhixian Yan. Semantic Trajectories Modeling and Analysis. *ACM Computing Surveys*, 45(4), August 2013. Article 42.

Streaming Analysis of Information Diffusion
Extended Abstract

Peter M. Fischer, Io Taxidou
Univ. of Freiburg, CS Department, 79110 Freiburg, Germany
{peter.fischer,taxidou}@informatik.uni-freiburg.de

1 Background and Motivation

Modern social media like Twitter or Facebook encompass a significant and growing share of the population, which is actively using it to create, share and exchange messages. This has a particularly profound effect on the way how news and events are spreading.Given the relevance of social media both as sensor of the real world (e.g., news detection) and its impact on the real world (e.g., shitstorms), there has been a significant work on fast, scalable and thorough analyses, with a special emphasis on trend detection, event detection and sentiment analysis.

To understand the *relevance* and *trustworthiness* of social media messages deeper insights into *Information Diffusion* are needed: where and by whom a particular piece of information has been created, how it has been propagated and whom it may have influenced.

Information diffusion has been a very active field of research, as recently described in a SIGMOD Record survey [GHFZ13]. The focus has been on developing models of information diffusion and targeted empirical studies. Given the complexity of most of these models, nearly all of the investigations have been performed on relatively small data sets, offline and in ad-hoc setups. Despite all this work, there is little effort to tackle the problem of real-time evaluation of information diffusion, which is needed to assess the relevance.

These analyses need to deal with *Volume* and *Velocity* on both on messages and social graphs. The combination of message streams and social graphs is scarely investigated, while incomplete data and complex models make reliable results hard to achieve. Existing systems do not handle the challenges: Graph computation systems neither address the fast change rates nor the realtime interaction between the graph and other processing, while data streams systems fall short on the combination of streams and complex graphs.

2 Goals and Challenges

The goal of our research is to develop algorithms and systems to trace the spreading of information in social media that produce large scale, rapid data. We identified three crucial building blocks for such a real-time tracing system:

1) Algorithms and Systems to perform the tracing and influence assignment, in order to deliver paths that information most likely propagated 2) Classification of user roles, as to provide support for assessing their impact on information diffusion process 3) Predictions on the spreading rate in order to allow estimations of information diffusion lifetime and how representative evaluations on the current state will be.

Our first task is to design, implement and evaluate algorithms and systems that can trace information spreading and assign influence at global scale while producing the results in real-time, matching the volumes and rates of social media. This requires a correlation between the message stream and the social graph. While we already showed that real-time reconstruction of retweets is feasible when the social graph fragment is locally accessible [TF14], real-life social graphs contain hundreds of millions of users, which requires *distributed storage and operation*. Our approach keeps track of the (past) interactions and drives the partitioning on the communities that exist in this interaction graph. Additionally, since the information available for reconstruction is incomplete, either from lack of social graph information or from API limitations, we aim to develop and evaluate methods that *infer missing path information* in a low-overhead manner. In contrast to existing, model-based approaches, we rely on a lightweight, neighborhood-based approach.

Access to diffusion paths enables a broad range of analyses of the information cascades. Given this broad range, we are specifically focusing on features that provide the baselines for supporting relevance and trustworthiness, namely the identification of prominent user roles such as opinion leaders or bridges. An important aspect includes the interactions and connections among users, that lead towards identifying prominent user roles. Our approach will rely on stream-aware clustering instead of fixed roles over limited data.

The process of information spreading varies significantly in speed and duration: most cascades end after a short period, others are quickly spreading for a short time, while yet other group see multiple peaks of activity or stay active for longer periods of time. Understanding how long such a diffusion continues provides important insights on how relevant a piece of information is and how complete its observation is. Virality predictions from the start of a cascade are hard to achieve, while incremental, lightweight forecasts are more feasible. New observations can then be used to update and extend this forecast, incorporating temporal as well as strucutural features.

References

[GHFZ13] Adrien Guille, Hakim Hacid, Cécile Favre, and Djamel A. Zighed. Information diffusion in online social networks: a survey. *SIGMOD Record*, 42(2):17–28, 2013.

[TF14] Io Taxidou and Peter M. Fischer. Online Analysis of Information Diffusion in Twitter. In *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '14 Companion, 2014.

Towards a Framework for Sensor-based Research and Development Platform for Critical, Socio-technical Systems

Henrik Surm[1], Daniela Nicklas[2]

[1]Department for Computing Science,
University Oldenburg, Oldenburg

henrik.surm@uni-oldenburg.de

[2]Faculty of Information Systems and
Applied Computer Sciences
University of Bamberg, Bamberg
dnicklas@acm.org

1 Motivation

The complexity of critical systems, in which failures can either endanger human life or cause drastic economic losses, has dramatically increased over the last decades. More and more critical systems envolve into so-called "socio-technical systems": humans are integrated by providing and assessing information and by making decisions in otherwise semi-autonomous systems. Such systems depend heavily on situational awareness, which is obtained by processing data from multiple, heterogenous sensors: raw sensor data is cleansed and filtered to obtain features or events, which are combined, enriched and interpreted to reach higher semantical levels, relevant for system decisions.

Existing systems which use sensor data fusion often require an a-priori configuration which can not be changed while the system is running, or require a human intervention to adapt to changed sensor sources, and provide no run-time-extensibility for new sensor types [HT08]. In addition, analysis of data quality or query plan reliability is often not possible and management of recorded data frequently is done by hand.

Our goal is to support the research, development, evaluation, and demonstration of such systems. Thus, in the proposed talk we analyze requirements and challenges for the data management of sensor based research environments, and we propose a data stream based architecture which fulfills these requirements.

2 Challenges and Requirements

To support the research, development, test, and demonstration of sensor-based, critical applications, we plan to address the following challenges and requirements:

- Changing sensor configurations: during research and development, the sensor configuration might change often. In addition, when sensor data is delivered from moving objects, the available sensor sources may change during runtime.
- Information quality: since sensors do not deliver exact data, quality aspects needs to be considered at all levels of processing.

- Validation of data management: since the sensor data processing is a vital part of the system, it should be validatable and deterministic.
- Reproducability of experiments and tests, intelligent archiving: in the process of research, development and test, application sensor-data and additional data for ground-truth (e.g., video-streams) needs to be archived and replayable.
- Integration with simulation environments: before such systems are deployed in the real, they need to be modeled and analyzed in various simulation tools. The transition from pure simulation to pure real-world execution should be easy.

In the project, we plan to address these challenges by a modular, extensible and comprehensive framework. While these challenges are similar to other data analysis and integration scenarios, they have to be addressed under the specific constraints of limited maritime communication channels and data standards.

The main contribution will be a combined solution that adresses these challenges in a unified framework. This is why we will base our work on the Odysseus [AGGM12] framework: it has a clear formal foundation based on [K08] and is designed for extensibility, using the OSGI service platform. Odysseus offers bundles for data integration, mining, storage and data quality representation, and allows installation and updates of bundles without restarting the system, which leads to a high flexibility and run-time adaptability. The framework will offer a unified sensor access and data fusion approach which allows a flexible use of the platform for the transition from simulation to real-world applications, reducing the expenditure of time for research and development.

3 Outlook

The framework will be used in two scenarios, which is a research port for analysis of sensor-based support for port navigation, and driver state observations for cooperative, semi-autonomous driving applications. Further, we plan to explore that approach in other projects, covering cooperative e-navigation and smart city applications.

References

[AGGM12] APPELRATH, H.-JÜRGEN ; GEESEN, DENNIS ; GRAWUNDER, MARCO ; MICHELSEN, TIMO ; NICKLAS, DANIELA: Odysseus: a highly customizable framework for creating efficient event stream management systems. In: *DEBS '12 Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, *DEBS '12*. New York, NY, USA : ACM, 2012 — ISBN 978-1-4503-1315-5, S. 367–368

[HT08] HE, YINGJIE ; TULLY, ALAN: Query processing for mobile wireless sensor networks: State-of-the-art and research challenges. In: *Third International Symposium on Wireless Pervasive Computing (ISWPC '08),* S. 518 - 523. IEEE Computer Society 2008 — ISBN 978-1-4244-1652-3

[K07] KRÄMER, JÜRGEN: Continuous Queries over Data Streams - Semantics and Implementation. Dissertation, Universität Marburg, 2007

Dataflow Programming for Big Engineering Data – extended abstract –

Felix Beier, Kai-Uwe Sattler, Christoph Dinh, Daniel Baumgarten
Technische Universität Ilmenau, Germany
{first.last}@tu-ilmenau.de

Nowadays, advanced sensing technologies are used in many scientific and engineering disciplines, e. g., in medical or industrial applications, enabling the usage of data-driven techniques to derive models. Measures are collected, filtered, aggregated, and processed in a complex analytic pipeline, joining them with static models to perform high-level tasks like machine learning. Final results are usually visualized for gaining insights directly from the data which in turn can be used to adapt the processes and their analyses iteratively to refine knowledge further. This task is supported by tools like R or MATLAB, allowing to quickly develop analytic pipelines. However, they offer limited capabilities of processing very large data sets that require data management and processing in distributed environments – tasks that have vastly been analyzed in the context of database and data stream management systems. Albeit the latter provide very good abstraction layers for data storage, processing, and underlying hardware, they require a complex setup, provide only limited extensibility, and hence are hardly used in scientific or engineering applications [ABB+12]. As consequence, many tools are developed, comprising optimized algorithms for specialized tasks, but burdening developers with the implementation of low-level data management code, usually in a language that is not common in their community.

In this context, we analyzed the source localization problem for EEG/MEG signals (which can be used, e. g., to develop therapies for stroke patients) in order to develop an approach for bridging this gap between engineering applications and large-scale data management systems. The source localization problem is challenging, since the problem is ill-posed and signal-to-noise ratio (SNR) is very low. Another challenging problem is the computational complexity of inverse algorithms. While large data volumes (brain models and high sampling rates) need to be processed, low latencies constraints must be kept because interactions with the probands are necessary. The analytic processing chain is illustrated in Fig. 1. The Recursively Applied and Projected Multiple Signal Classification (RAP-MUSIC) algorithm is used for locating neural sources, i. e., activity inside a brain corresponding to a specific input. Therefore, 366 MEG/EEG sensors are placed above the head which are continuously sampled at rates of 600 – 1250 Hz. The forward solution of the boundary element model (BEM) of the brain at uniformly distributed locations on the white matter surface is passed as second input. It is constructed once from a magnetic resonance imaging (MRI) scan and, depending on the requested accuracy, comprises 10s of thousands of vertices representing different locations on the surface. RAP-MUSIC recursively identifies active neural regions with a complex pipeline for preprocessing signal measures and correlating them with the BEM. To meet the latency constraints, the RAP-

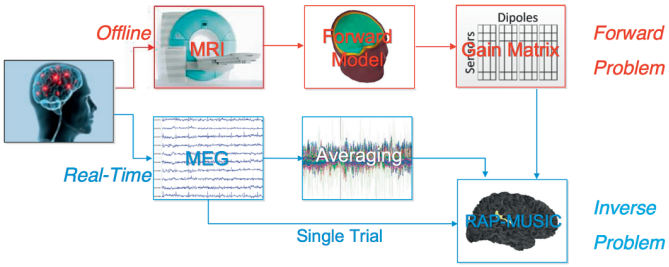


Figure 1: Overview Source Localization Processing Chain

MUSIC algorithm has been parallelized for GPUs and a C++ library has been created in an analysis tool called MNE-CPP [DLS+13], including parsers for data formats used by vendors of medical sensing-equipment, signal filter operators, transformation routines, etc. Although this library can be used to create larger analyses pipelines, implementing and evaluating new algorithms still requires a lot of low-level boilerplate code to be written, leading to significant development overheads. Latter can be avoided with the usage of domain-specific languages which are specialized for signal processing, natively working on vectors or matrices as first-class data types like MATLAB. But they offer less control over memory management and parallelization for custom algorithms which are crucial for meeting the latency constraints. When large-scale data sets shall be processed, even a specialized tool quickly runs into performance problems as distributed processing is mostly not supported because of large development overheads for cluster-scale algorithms.

To handle these problems, we propose to apply dataflow programming here. We implemented a multi-layered framework which allows to define analytic programs by an abstract flow of data, independent from its actual execution. This enables quick prototyping while letting the framework handle data management and parallelization issues. Similar to Pig for batch-oriented MapReduce jobs, a scripting language for stream-oriented processing, called *PipeFlow*, is provided as front-end. In the current version, *PipeFlow* allows to inject primitives for partitioning dataflows, executing sub-flows in parallel on cluster nodes leveraging multi-core CPUs, and merging partial results. We plan to automatically parallelize flows in the future using static code analysis and a rule-based framework for exploiting domain-specific knowledge about data processing operators. The dataflow programs are optimized by applying graph rewriting rules and code for an underlying execution back-end is generated. Therefore, the framework provides an engine called *PipeFabric* which offers a large C++ library of operator implementations with focus on low-latency processing. One key aspect of *PipeFabric* is its extensibility for complex user-defined types and operations. Simple wrappers are sufficient to embed already existing domain-specific libraries. For our use case, processing of large matrices is required. Therefore, we used the Eigen library and are currently porting functions from MNE-CPP. We are also working on code generators for other back-ends like Spark which will be useful for comparing capabilities of different frameworks for common analytic workloads – which, to the best of our knowledge, has not been done yet.

References

[ABB+12] I. Alagiannis, R. Borovica, M. Branco, S. Idreos, and A. Ailamaki. NoDB: efficient query execution on raw data files. In *ACM SIGMOD*, 2012.

[DLS+13] C. Dinh, M. Luessi, L. Sun, J. Haueisen, and M. S Hamalainen. Mne-X: MEG/EEG Real-Time Acquisition, Real-Time Processing, and Real-Time Source Localization Framework. *Biomedical Engineering/Biomedizinische Technik*, 2013.

Joint Workshop on Data Management for Science

Joint Workshop on Data Management for Science

Sebastian Dorok[1,5], Birgitta König-Ries[2], Matthias Lange[3],
Erhard Rahm[4], Gunter Saake[5], Bernhard Seeger[6]

[1] Bayer Pharma AG
[2] Friedrich Schiller University Jena
[3] Leibniz Institute of Plant Genetics and Crop Plant Research (IPK) Gatersleben
[4] University of Leipzig
[5] Otto von Guericke University Magdeburg
[6] Philipps University Marburg

Message from the chairs

The *Workshop on Data Management for Science* (DMS) is a joint workshop consisting of the two workshops *Data Management for Life Sciences* (DMforLS) and *Big Data in Science* (BigDS). BigDS focuses on addressing big data challenges in various scientific disciplines. In this context, DMforLS focuses especially on life sciences. In the following, we give short excerpts of the call for papers of both workshops:

Data Management for Life Sciences In life sciences, scientists collect an increasing amount of data that must be stored, integrated, processed, and analyzed efficiently to make effective use of them. Thereby, not only the huge volume of available data raises challenges regarding storage space and analysis throughput, but also data quality issues, incomplete semantic annotation, long term preservation, data access, and compliance issues, such as data provenance, make it hard to handle life science data. To address these challenges, advanced data management techniques and standards are required. Otherwise, the use of life science data will be limited. Thereby, one question is whether general purpose techniques and methods for data management are suitable for life science use cases or specialized solutions tailored to life science applications must be developed.

Big Data in Science The volume and diversity of available data has dramatically increased in almost all scientific disciplines over the last decade, e.g. in meteorology, genomics, complex physics simulations and biological and environmental research. This development is due to great advances in data acquisition (e.g. improvements in remote sensing) and data accessibility. On the one hand, the availability of such data masses leads to a rethinking in scientific disciplines on how to extract useful information and on how to foster research. On the other hand, researchers feel lost in the data masses because appropriate data management tools have been not available so far. However, this is starting to change with the recent development of big data technologies that seem to be not only useful in business, but also offer great opportunities in science.

The joint workshop DMS brings together database researchers with scientists from various disciplines especially life sciences to discuss current findings, challenges, and opportunities of applying data management techniques and methods in data-intensive sciences. The joint workshop is held for the first time in conjunction with the 16th Conference on Database Systems, Technology, and Web (BTW 2015) at the University of Hamburg on March 03, 2015.

The contributions were reviewed by three to four members of the respective program committee. Based on the reviews, we selected eight contributions for presentation at the joint workshop. We assigned each contribution to one of three different sessions covering different main topics.

The first session comprises contributions related to *information retrieval*. The contribution *Ontology-based retrieval of scientific data in LIFE* by Uciteli and Kirsten presents an approach that utilizes ontologies to facilitate query formulation. Colmsee et al. make also use of ontologies, but use them for improving search results. In *Improving search results in life science by recommendations based on semantic information*, they describe and evaluate their approach that uses document similarities based on semantic information. To improve performance of sampling analyses using MapReduce, Schäfer et al. present an incremental approach. In *Sampling with incremental MapReduce*, the authors describe a way to limit data processing to updated data.

In the next session, we consolidate contributions dealing with *data provenance*. In his position paper *METIS in PArADISE*, Heuer examines the importance of data provenance in the evaluation of sensor data, especially in assistance systems. In their contribution *Extracting reproducible simulation studies from model repositories using the COMBINE archive toolkit*, Scharm and Waltemath deal with reproducible simulation studies.

The last session covers the topic *data analysis*. In *Genome sequence analysis with MonetDB: a case study on Ebola virus diversity*, Cijvat et al. present a case study on genome analysis using a relational main-memory database system as platform. In *RightInsight: Open source architecture for data science*, Bulut presents an approach based on Apache Spark to conduct general data analyses. In contrast, Authmann et al. focus on challenges in spatial applications and suggest an architecture to address them in their paper *Rethinking spatial processing in data-intensive science*.

We are deeply grateful to everyone who made this workshop possible – the authors, the reviewers, the BTW team, and all participants.

Program chairs

Data Management for Life Sciences
Gunter Saake (Otto von Guericke University Magdeburg)
Uwe Scholz (IPK Gatersleben)

Big Data in Science
Birgitta König-Ries (Friedrich Schiller University Jena)
Erhard Rahm (University of Leipzig)
Bernhard Seeger (Philipps University Marburg)

Program committee

Data Management for Life Sciences
Sebastian Breß (TU Dortmund)
Sebastian Dorok (Otto von Guericke University Magdeburg)
Mourad Elloumi (University of Tunis El Manar, Tunisia)
Ralf Hofestädt (Bielefeld University)
Andreas Keller (Saarland University, University Hospital)
Jacob Köhler (DOW AgroSciences, USA)
Matthias Lange (IPK Gatersleben)
Horstfried Läpple (Bayer HealthCare AG)
Ulf Leser (Humboldt-Universität zu Berlin)
Wolfgang Müller (HITS GmbH)
Erhard Rahm (University of Leipzig)
Can Türker (ETH Zürich, Switzerland)

Big Data in Science
Alsayed Algergawy (Friedrich Schiller University Jena)
Peter Baumann (Jacobs Universität)
Matthias Bräger (CERN)
Thomas Brinkhoff (FH Oldenburg)
Michael Diepenbroeck (Alfred-Wegner-Institut)
Christoph Freytag (Humboldt Universität)
Michael Gertz (Uni Heidelberg)
Frank-Oliver Glöckner (MPI für Marine Mikrobiologie)
Anton Güntsch (Botanischer Garten und Botanisches Museum, Berlin-Dahlem)
Thomas Heinis (Imperial College, London)
Thomas Hickler (Senckenberg)
Jens Kattge (MPI für Biogeochemie)
Alfons Kemper (TU München)
Meike Klettke (Uni Rostock)
Alex Markowetz (Uni Bonn)
Thomas Nauss (Uni Marburg)
Jens Nieschulze (Forschungsreferat für Datenmanagement der Uni Göttingen)
Kai-Uwe Sattler (TU Ilmenau)
Stefanie Scherzinger (OTH Regensburg)
Myra Spiliopoulou (Uni Magdeburg)
Uta Störl (HS Darmstadt)

Ontology-based Retrieval of Scientific Data in LIFE

Alexandr Uciteli[1,2], Toralf Kirsten[2,3]

[1]Institute for Medical Informatics, Statistics and Epidemiology, University of Leipzig
[2]LIFE Research Centre for Civilization Diseases, University of Leipzig
[3]Interdisciplinary Centre for Bioinformatics, University of Leipzig

Abstract: LIFE is an epidemiological study determining thousands of Leipzig inhabitants with a wide spectrum of interviews, questionnaires, and medical investigations. The heterogeneous data are centrally integrated into a research database and are analyzed by specific analysis projects. To semantically describe the large set of data, we have developed an ontological framework. Applicants of analysis projects and other interested people can use the LIFE Investigation Ontology (LIO) as central part of the framework to get insights, which kind of data is collected in LIFE. Moreover, we use the framework to generate queries over the collected scientific data in order to retrieve data as requested by each analysis project. A query generator transforms the ontological specifications using LIO to database queries which are implemented as project-specific database views. Since the requested data is typically complex, a manual query specification would be very time-consuming, error-prone, and is, therefore, unsuitable in this large project. We present the approach, overview LIO and show query formulation and transformation. Our approach runs in production mode for two years in LIFE.

1 Introduction

Epidemiological projects study the distribution, the causes and the consequences of health-related states and events in defined populations. The goal of such projects is to identify risk factors of (selected) diseases in order to establish and to optimize a preventive healthcare. LIFE is an epidemiological and multi-cohort study in the described context at the Leipzig Research Centre for Civilization Diseases (Univ. of Leipzig). The goal of LIFE is to determine the prevalence and causes of common civilization diseases including adiposity, depression, and dementia by examining thousands of Leipzig (Germany) inhabitants of different ages. Participants include pregnant women and children from 0-18 as well as adults in separate cohorts. All participants are determined in a possible multi-day program with a selection out of currently more than 700 assessments. To these assessments belong interviews and self-completed questionnaires to physical examinations, such as for anthropometry, EKG, MRT, and laboratory analyses of taken specimen. Data is acquired for each assessment depending on the participant's investigation program using specific input systems and prepared input forms. All collected data is integrated and, thus, harmonized in a central research database. This database consists of data tables referring to assessments (i.e., investigations). Their complexity ranges from tables with a small number of columns to tables with a very high column number. For example, the table referring to the Structured Clinical Interview (SCID) consists of more than 900 columns, i.e., questions and sub-questions of the interview input form.

The collected data are analyzed in an increasing number of analysis projects; currently, there are more than 170 projects active. Each project is initially specified by a proposal

documenting the analysis goal, plan and the required data. However, there are two key aspects that are challenging. Firstly, the applicant needs to find assessments (research database tables) of interest to specify the requested data in the project proposal. This process can be very difficult and time consuming, in particular, when the scientist is looking for specific data items (columns), such as weight and height, without knowing the corresponding assessment. Secondly, current project proposals typically request data from up to 50 assessments which are then organized in project-specific views (according to the data requests). These views can be very complex. Usually, they combine data from multiple research database tables, several selection expressions and a multitude on projected columns out of the data tables. A manual specification of database queries to create such views for each analysis project would be a very error-prone and time-consuming process and is, therefore, nearly impossible. Hence, we make the following contributions.

- We developed an ontological framework. The framework utilizes the LIFE Investigation Ontology (LIO) which classifies and describes assessments, relations between them, and their items.
- We implemented ontology-based tools using LIO to generate database queries which are stored as project-specific analysis views within the central research database. The views allow scientists and us to easily access and to export the requested data of an analysis project. Both, LIO and ontology-based tools are running in production mode for two years.

The rest of the paper is organized as follows. The Section 2 describes the ontological framework and especially LIO. The Section 3 deals with ontology-based query formulation and transformation, while Section 4 describes some implementation aspects. Section 5 concludes the paper.

2 Framework

The goal of the ontological framework is to semantically describe all integrated data of biomedical investigations in LIFE using an ontology. The ontology is utilized on the one hand by scientists to search for data items or complete investigations of interest or simply to browse the ontology to get information about the captured data of the investigations. On the other hand, the ontology helps to query and retrieve data of the research database by formulating queries on a much higher level than SQL.

The ontological framework consists of three interrelated layers (Fig. 1). The integrated data layer comprises all data elements (instance data) of the central research database providing data of several source systems in an integrated, preprocessed and cleaned fashion. The metadata layer describes all instance data of the research database on a very technical level. To these metadata belongs the used table and column names, corresponding data types but also the original question or measurement text and the code list when a predefined answer set has been originally associated to the data item. This metadata is stored in a dedicated metadata repository (MDR) and is inherently interrelated with

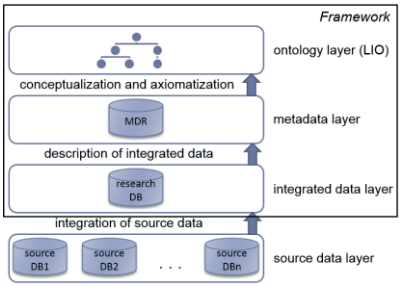


Figure 1: Framework overview

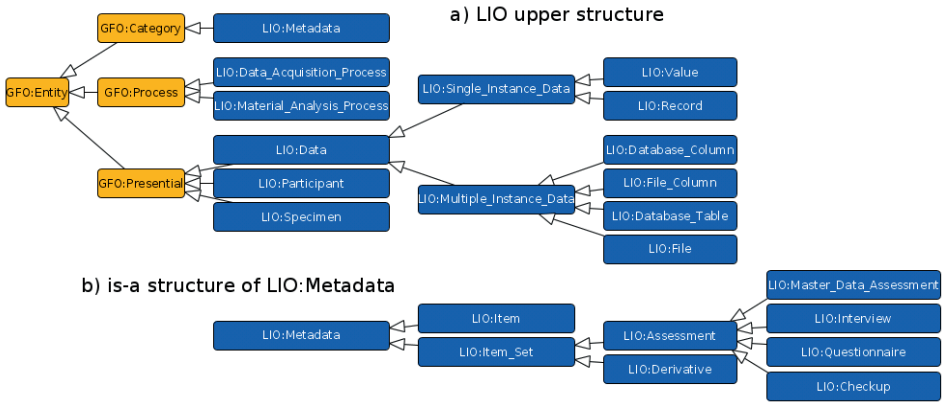


Figure 2: Selection of the LIFE Investigation Ontology

the instance data. Finally, the ontology layer is represented by the developed LIFE Investigation Ontology (LIO) [KK10] and its mapping to the collected metadata in the MDR.

LIO utilizes the General Formal Ontology (GFO) [He10] as a top-level ontology and, thus, reuses defined fundamental categories of GFO, such as *Category*, *Presential* and *Process*. Fig. 2a gives a high-level overview over LIO. Subcategories of *GFO:Presential* refer to collected scientific data, participants and specimen. Scientific data is structured on a technically level by categories within the sub-tree of *LIO:Data*. Instances of these categories are concrete data files and database tables, e.g., of the research database. They are used to locate instance data for later querying. Subcategories of *GFO:Process* refer to processes of two different types, data acquisition and material analysis processes. Instances of these categories are documented, e.g., by specific states and conditions of the examination, the examiner conducting the investigation etc. This process documentation is additionally specified to the scientific data that they possibly generate. The documentation can be used for downstream analysis of the scientific data, to evaluate the process quality and for an impact analysis on the measurement process. Finally, subcategories of *GFO:Category* are utilized to semantically classify biomedical investigations in LIFE. Fig. 2b shows an overview of main categories. Fundamentally, we differentiate between items (e.g., questions of a questionnaire) and item sets. This separation allows us to ontologically distinguish between data tables (item set) of the research database and its columns (item). Moreover, we are able to classify both, investigation forms as predefined and rather static item sets and specific project-specific analysis views which potentially include items from multiple investigation forms. The latter can be dynamically defined and provided by a user group.

All biomedical investigations together with their containing items (i.e., questions and measurements) are associated to LIO categories using the *instance_of* relationship type. Fig. 3 shows an example; the interview socio-demography consists of several questions including for country of birth, material status and graduation. Both, the interview and its items, are associated to *LIO:Interview* and *LIO:Item*, respectively. Special relationships with type *has_item* represent the internal structure of the interview. The semantic classification, i.e., the association to LIO subclasses of *LIO:Metadata*, is manually specified by the investigator in an operational software application from which it is imported into

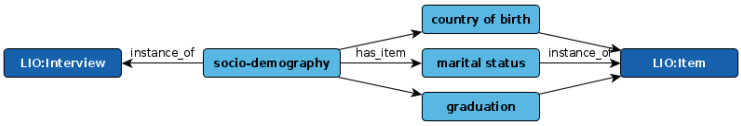


Figure 3: Utilization of item set and item categories to describe investigations

the overall metadata repository (MDR). Moreover, the MDR captures and manages the structure of each investigation form, and, thus, its items, and their representation in the central research database. By reusing both kinds of specifications in LIO, the mappings between collected metadata in the MDR and LIO categories are inherently generated. This makes it easy to describe and classify new investigations (assessments) in LIO; it necessitates only an initial manual semantic classification and import of corresponding metadata into the MDR.

3 Ontology-based Query Formulation and Transformation

Scientific data of the central research database are analyzed in specific analysis projects. Each of them is specified by a project proposal, i.e., the applicant describe the analysis goal, the analysis plan and the data she request for. In the simplest case, the data request consists in a list of assessments. This is extended, in some cases, by defined inclusion or exclusion criteria. In complex scenarios, the applicant is interested in specific items instead of all items of an assessment. In all scenarios, data can be queried per single assessment. However, it is common to request data in a joined fashion, especially, when the applicant focuses on specific items from multiple assessments. Currently, each data request is satisfied by specific analysis views which are implemented as database views within the relational research database.

We use LIO to formulate queries over the scientific data which are finally transformed and stored as project-specific analysis views in the research database. Fig. 4 sketches the query formulation and transformation process. Firstly, LIO is used to formulate queries for each analysis project. The applicant can search for assessments of interest browsing along LIO's structure and the associated instances, i.e., concrete assessments or items. She can select complete assessments as predefined item sets and specific items of an assessment. These selections are used by the query generator to create the query projection, i.e., the items for that data should be retrieved. These items are firstly sorted by the selected assessment in alphabetic order and, secondly, by their rank within each assessment, i.e., with respect to their position on the corresponding input form. Inclusion and exclusion criteria can be specified on item level. The query generator interrelates single conditions by the logical operator AND and creates the selection expression of the resulting query for each assessment.

Per default, the query generator produces one query for each selected assessment or item set of an assessment. Moreover, experienced users can create new item sets containing items from multiple assessments. These item sets result in join queries using patient identifiers and examination time points (due to recurrent visits) as join criteria. To find out which item (column) contains patient identifiers and time points, the items are specifically labelled when the assessment definitions (source schema) are imported into the MDR. Some sources allow an automatic labelling (using source-specific rules),

whereas other sources need a manual intervention to fully describe their schemas and the resulting items of LIO.

The query generator takes the ontology-based specifications (selections and conditions) using LIO as input and firstly creates SQL-like queries. These queries have intermediate character and utilize keywords SELECT, FROM and WHERE with the same meaning as in SQL. In contrast to SQL, they contain ontology categories and associated instances as placeholder which are then finally replaced by conrete table and column names of the research database when SQL queries are generated. To resolve table and column names the query generator utilizes mappings between LIO and the MDR.

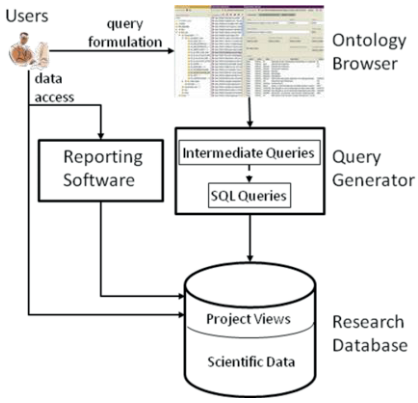


Figure 4: Query fromulation and data access

4 Implementation

LIO currently consists of 33 categories, more than 700 assessments and ca. 120 analysis results (latter two are instances in LIO) together with more than 39,000 items in total. The large and increasing number of assessments, their containing items and their correspondences (mappings) to database table and column metadata are stored in the MDR which is implemented in a relational database system. Assessments and items are loaded on demand from the MDR and are associated to LIO categories as instances. Therefore, new assessments can be easily added to LIO and without modifying the LIO's core structure or changing ontology files.

We implemented a Protégé [NFM00, Sc11] plug-in loading LIO and corresponding instances from the MDR to support an applicant when she specifies the required data for a project proposal. She can navigate along LIO's structure and, hence, is able to find and pick the items of interest for her proposal. On the other hand, the plug-in allows us to formulate and to transfer ontology-based queries into SQL-queries which are then stored as project-specific analysis views over the scientific data of the research database. These views can be access in two different ways. Firstly, the views can be used for further database-internal data processing using the database API and SQL. This is the most preferred way for persons with database skills. Secondly, the plug-in includes options to propagate views to a web-based reporting software which wraps the database views in tabular reports. These reports can be executed by an applicant. The retrieved data are then available for download to continue data processing with special analysis tools, such as SPSS, R etc.

There are other approaches which are highly related to our ontology-based framework. i2b2 [Mu10] is a framework for analyzing data in clinical context. In contrast to our approach, it utilizes a separate data management and, thus, necessitates additional data load and transformation processes. Moreover, the goal of i2b2 is primarily to find relevant patients and not to retrieve scientific data. Like LIO, the Search Ontology [Uc14] is used to formulate queries over data. Its focus is on queries for search engines, while our

approach focuses on structured data in a relational database. Similar to LIO, the Ontology of Biomedical Investigations (OBI) [Br10] classifies and describes biomedical investigations. In contrast to OBI, LIO utilizes a core structure which is dynamically extended by assessments fully described in a dedicated metadata repository. Hence, our framework is able to generate queries over data of the research database and prevents from describing each investigation in detail by using OBI.

5 Conclusion

We introduced an ontology-based framework to query large and heterogeneous sets of scientific data. The framework consists of the developed LIFE Investigation Ontology (LIO) on the top level which semantically describes scientific data of the central research database (base level). Both levels are interrelated by (technical) metadata which are managed in a metadata repository. LIO gets insight which data are available within the research database, on the one hand, and is used, on the other hand, to formulate queries over the collected scientific data. The ontology-based queries are transformed into database queries which are stored as analysis-specific database views. The queries include per default items of a single assessment. Moreover, join queries merging items from multiple assessments are also supported. Together, ontology-based querying simplifies the data querying for end users and frees IT-people from implementing rather complex SQL queries. In future, we will extend LIO and the query generator to overcome current limitations, e.g., according to the specification and transformation of query conditions.

Acknowledgment: This work was supported by the LIFE project. The research project is funded by financial means of the European Union and of the Free State of Saxony. LIFE is the largest scientific project of the Saxon excellence initiative.

References

[Br10] Brinkman, R. R. et al.: Modeling biomedical experimental processes with OBI. In Journal of biomedical semantics, 2010, 1 Suppl 1; pp. S7.

[He10] Herre, H.: General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling. In (Poli, R.; Healy, M.; Kameas, A. Eds.): Theory and Applications of Ontology: Computer Applications. Springer Netherlands, Dordrecht, 2010; pp. 297–345.

[KK10] Kirsten, T.; Kiel, A.: Ontology-based Registration of Entities for Data Integration in Large Biomedical Research Projects. In (Fähnrich, K.-P.; Franczyk, B. Eds.): Proceedings of the annual meeting of the GI. Köllen Druck+Verlag GmbH, Bonn, 2010; pp. 711–720.

[Mu10] Murphy, S. N. et al.: Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). In Journal of the American Medical Informatics Association JAMIA, 2010, 17; pp. 124–130.

[NFM00] Noy, N. F.; Fergerson, R. W.; Musen, M. A.: The Knowledge Model of Protégé-2000: Combining Interoperability and Flexibility. In (Goos, G. et al. Eds.): Knowledge Engineering and Knowledge Management Methods, Models, and Tools. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000; pp. 17–32.

[Sc11] Schalkoff, R. J.: Protégé, OO-Based Ontologies, CLIPS, and COOL: Intelligent systems: Principles, paradigms, and pragmatics. Jones and Bartlett Publishers, Sudbury, Mass., 2011; pp. 266–272.

[Uc14] Uciteli, A. et al.: Search Ontology, a new approach towards Semantic Search. In (Plödereder, E. et al. Eds.): FoRESEE: Future Search Engines 2014 - 44. annual meeting of the GI, Stuttgart - GI Edition Proceedings P-232. Köllen, Bonn, 2014; pp. 667–672.

Improving Search Results in Life Science by Recommendations based on Semantic Information

Christian Colmsee[1], Jinbo Chen[1], Kerstin Schneider[2], Uwe Scholz[1], Matthias Lange[1]

[1]Department of Cytogenetics and Genome Analysis
Leibniz Institute of Plant Genetics and Crop Plant Research (IPK) Gatersleben
Corrensstr. 3
06466 Stadt Seeland, Germany
{colmsee,chenj,scholz,lange}@ipk-gatersleben.de

[2]Department Automation / Computer Science
Harz University of Applied Sciences
Friedrichstr. 57-59
38855 Wernigerode, Germany
{kschneider}@hs-harz.de

Abstract: The management and handling of big data is a major challenge in the area of life science. Beside the data storage, information retrieval methods have to be adapted to huge data amounts as well. Therefore we present an approach to improve search results in life science by recommendations based on semantic information. In detail we determine relationships between documents by searching for shared database IDs as well as ontology identifiers. We have established a pipeline based on Hadoop allowing a distributed computation of large amounts of textual data. A comparison with the widely used cosine similarity has been performed. Its results are presented in this work as well.

1 Introduction

Nowadays the management and handling of big data is a major challenge in the field of informatics. Larger datasets are produced in less time. In particular, this aspect is intensively discussed in life science. At a technology level, new concepts and algorithms have to be developed to enable a seamless processing of huge data amounts. In the area of data storage new database concepts are implemented, such as column based storage or in memory databases. In respect to data processing, distributed data storage and computation have been made available by new frameworks such as the Hadoop framework (http://hadoop.apache.org). Hadoop is using the MapReduce approach [DG08] allowing the distribution of tasks in the map phase over different clusters and to reduce the amount of data in the reduce phase. Furthermore Hadoop is able to integrate extensions such as the column oriented database HBase. So the framework combines the distributed computation architecture with the advantages of a NoSQL database system. Hadoop has already been used in life science applications such as Hadoop-BAM [NKS+12] and Crossbow [LSL+09].

Beside these technological aspects, information retrieval (IR) plays an important role as well. In this context search engines play a pivotal role for an integrative IR over widely spread and heterogeneous biological data. Search engines are complex software systems and have to fulfil various qualitative requirements to get accepted by the scientific community. Its major components are discussed in [LHM+14]:

- Linguistic (text and data decomposition, e.g. tokenization; language processing, e.g. stop words and synonyms)
- Indexing (efficient search, e.g. inverse text index)
- Query processing (fuzzy matching and query expansion, e.g. phonetic search, query suggestion, spelling correction)
- Presentation (intuitive user interface, e.g. faceted search)
- Relevance estimation (feature extraction and ranking, e.g. text statistics, text feature scoring and user pertinence)
- Recommender systems (semantic links between related documents, e.g. "page like this" and "did you mean"

The implementation of those components is part of the research project LAILAPS [ECC+14]. LAILAPS is an information retrieval system to link plant genomic data in the context of phenotypic attributes for a detailed forward genetic research. The underlying search engine allows fuzzy querying for candidate genes linked to specific traits over a loosely integrated system of indexed and interlinked genome databases. Query assistance and an evidence based annotation system enable a time efficient and comprehensive information retrieval. The results are sorted by relevance using an artificial neural network incorporating user feedback and behaviour tracking.

While the ranking algorithm of LAILAPS provides user specific results, the user might be interested in links to other relevant database entries to an entry of his interest. Such a recommender system is still a missing LAILAPS feature but would have enormous impact for the quality of search results. A scientist may search for a specific gene to retrieve all relevant information to this gene without a dedicated search in different databases. To realise such a goal, recommendation systems are a widely used method in information retrieval. This concept is already used in several life science applications. For example, EB-eye as IR system for all databases that are hosted at the European Bioinformatics Institute (EBI) provide suggestions to alternative database records [VSG+10]. Another example are PubMed based IR systems for searching in biomedical abstracts [Lu11]. In this work we will describe a concept of providing recommendations in LAILAPS based on semantic information.

2 Results

When users are searching in LAILAPS for specific terms, the result is a list of relevant database entries. Beside the particular search result, the user would benefit from a list of related database entries that are potentially of interest to him. To implement this feature it is necessary to measure the similarity between database entries. A widely used concept is the expression of a document as a vector of words (tokens) and the computation of its

distance by cosine similarity. Within this approach the tokens of each document will be compared, meaning documents using similar words have a higher similarity to each other. But to get a more useful result especially in the context of life science it would be necessary to integrate semantic information in the comparison of documents. Here we present a method allowing the estimation of semantic relationships of documents.

2.1 Get semantics with database identifiers

A widely used concept to provide semantic annotation in life science databases are ontologies such as the Gene Ontology (GO) [HCI+04]. Each GO term has a specific ID allowing the exact identification of a term. Beside the use of ontologies, annotation targets are repositories of gene functions. This wide range of databases such as Uniprot [BAW+05], have in common that they can be referenced by a unique identifier for each database entry.

With the help of such unique identifiers for ontologies and database entries the documents in LAILAPS can be compared on a semantic level. If for example two documents share a GO identifier, this could be interpreted as a semantic connection between these documents. The final goal therefore would be to design a recommendation system, which is determining these information and to recommend the end user database entries based on these unique identifier.

For the extraction of above mentioned identifiers, different methods could be applied. One method is the usage of regular expressions, where specific patterns are used, such as a token beginning with the letters GO, is likely a GO term [BSL+11]. Another method is described in Mehlhorn et al. [MLSS12], where predictions are made with the support of a neural network. Feature extractions were focused on positions, symbols as well as word statistics to predict a database entry identifier. To include a very high number of database identifiers, we decided to use the neural network based approach, allowing the identification of IDs based on known ID patterns.

2.2 Determine document relations

We applied Hadoop to identify IDs in a high throughput manner. The Hadoop pipeline has two MapReduce components (see Figure 1). The first MapReduce job has a database as an input file. Each database entry consists of a unique document ID as well as document content. The mapper will then analyse each document and detect tokens that might be an ID with the IDPredictor tool from Mehlhorn et al. [MLSS12]. The reducer will then generate a list of pairs with a token and documents including this token. The second MapReduce job will then determine the document relations. Here the mapper will build pairs of documents having an ID in common. The reducer will finally count the number of shared IDs for each document pair. A high count of shared IDs means a high similarity between two documents. The source code of the pipeline is available at: http://dx.doi.org/10.5447/IPK/2014/18.

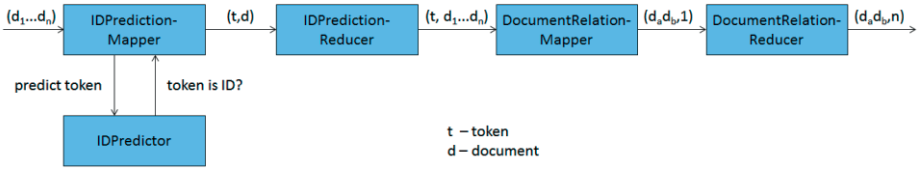


Figure 1: Hadoop pipeline including two MapReduce components as well as the ID prediction component

2.3 Cosine similarity versus ID prediction

As a benchmark we computed documents from the Swissprot database and compared the ranking results with the cosine similarity mentioned in section 2.1. While the cosine similarity score between two documents is built upon word frequencies and results in a value between zero and one, the ID prediction score is an integer value based on shared IDs. To make both values comparable, we calculated z-scores for both ranking scores. To detect deviations in the ranking, the results were plotted on a scatterplot (see Figure 2). The plot illustrates, that in most cases there are only small differences in the ranking. But there are some cases of large differences in the relative ranking, indicating, that for specific document relations the semantic component leads to a completely different ranking in contrast to the simple approach of comparing words.

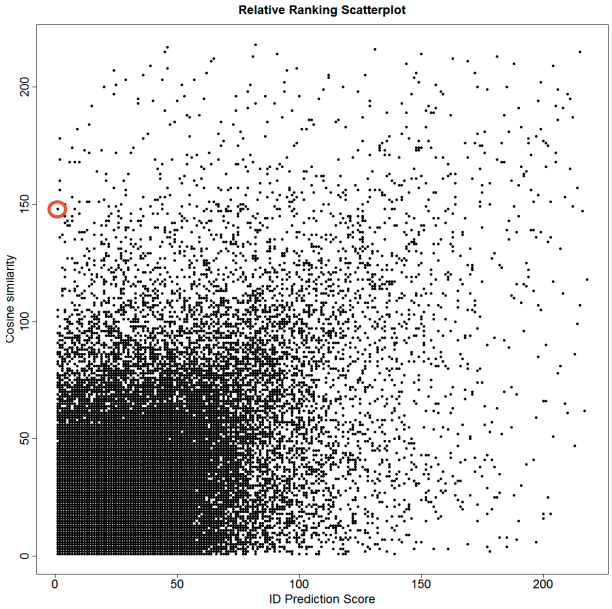


Figure 2: Scatterplot illustrating the different ranking results between cosine similarity and ID prediction score

When looking into specific results with strongly different rankings, semantically similarities could be detected. Picking up one example from Figure 2 (marked with a red circle) a document pair was ranked at place 1 in IDPrediction and at place 148 in cosine similarity. When looking into these documents we could determine that they are sharing a lot of IDs like EC (Enzyme Commission) numbers and GO terms. Both documents are dealing with fatty acid synthase in fungal species. A protein BLAST against Swissprot of the protein sequence of document A listed document B in the fourth position with a score of 1801 and an identity of 44%.

3 Discussion and Conclusion

In this work we developed a system providing recommendations based on semantic information. By the support of a neural network, IDs were predicted. With this information, documents can be compared on a semantic level. To support big data in life science we implemented the documents distance computation as a Hadoop pipeline. The results of our approach have shown differences to cosine similarity in case of rankings. The ID prediction based approach is able to detect semantic similarities between documents and recommend this information to the users. However to get a precise idea about the quality improvement, the new method should be applied to the LAILAPS frontend system to determine if the users are more interested in this new information. To implement the presented pipeline into LAILAPS powerful systems such as ORACLE Big Data [Dj13] could be a solution. It allows supporting multiple data source including Hadoop, NoSQL as well as the ORACLE database itself. Although Hadoop is a powerful system, LAILAPS would also benefit from a more integrative approach like using in memory technology. Users who would like to install their own LAILAPS instance might be not able to set up their own Hadoop cluster. In memory systems might be able to allow as the just-in-time computation of the available data as well. LAILAPS would benefit from further investigations in this field.

Acknowledgements

The expertise and support of Steffen Flemming regarding the establishment and maintenance of the Hadoop cluster is gratefully acknowledged.

References

[BAW+05] Bairoch, A.; Apweiler, R.; Wu, C.H.; Barker, W.C.;Boeckmann, B.; Ferro, S.; Gasteiger, E.; Huang, H.; Lopez, R.; Magrane, M. et al. The universal protein resource (Uniprot). Nucleic acids research, 33 (suppl 1):D154-D159, 2005.

[BSL+11] Bachmann, A.; Schult, R.; Lange, M.; Spiliopoulou, M. Extracting Cross References from Life Science Databases for Search Result

Ranking. In Proceedings of the 20th ACM Conference on Information and Knowledge Management, 2011.

[DG08] Dean, J.; Ghemawat, S.: Mapreduce: simpified data processing on large clusters. Communications of the ACM, 51(1):107-113, 2008.

[Dj13] Djicks, J.: Oracle: Big Data for the Enterprise. Oracle White Paper, 2013.

[ECC+14] Esch, M.; Chen, J.; Colmsee, C.; Klapperstück, M.; Grafahrend-Belau, E.; Scholz, U.; Lange, M.: LAILAPS – The Plant Science Search Engine. Plant and Cell Physiology, Epub ahead of print, 2014.

[HCI+04] Harris, M.A.; Clark, J.; Ireland, A.; Lomax, J.; Ashburner, M.; Foulger, R.; Eilbeck, K.; Lewis, S.; Marshall, B.; Mungall, C. et al.: The gene ontology (GO) database and informatics resource. Nucleic acids research, 32 (Database issue):D258, 2004.

[LHM+14] Lange, M; Henkel, R; Müller, W; Waltemath, D; Weise, S: Information Retrieval in Life Sciences: A Programmatic Survey. In M. Chen, R. Hofestädt (editors) Approaches in Integrative Bioinformatics. Springer, 2014, pp 73-109.

[LSL+09] Langmead, B.; Schatz, M.C.; Lin, J.; Pop, M.; Salzberg, S.L.: Searching for SNPs with cloud computing. Genome Biology, 10(11):R134, 2009.

[Lu11] Lu, Z.: PubMed and beyond: a survey of web tools for searching biomedical literature. Database. Oxford University Press, 2011

[MLSS12] Mehlhorn, H.; Lange, M.; Scholz, U.; Schreiber, F.: IDPredictor: predict database links in biomedical database. J. Integrative Bioinformatics, 9, 2012.

[NKS+12] Niemenmaa, M.; Kallio, A.; Schumacher, A.; Klemelä, P.; Korpelainen, E.; Heljanko, K.: Hadoop-BAM: directly manipulating next generation sequencing data in the cloud. Bioinformatics, 28(6):876-877, 2012.

[VSG+10] Valentin, F.; Squizzato, S.; Goujon, M.; McWilliam, H.; Paern, J.; Lopez, R.: Fast and efficient searching of biological data resources - using EB-eye. Briefings in bioinformatics, 11(4):375-384, 2010.

Sampling with Incremental MapReduce

Marc Schäfer, Johannes Schildgen, Stefan Deßloch

Heterogeneous Information Systems Group
Department of Computer Science
University of Kaiserslautern
D-67653 Kaiserslautern, Germany
{m_schaef,schildgen,dessloch}@cs.uni-kl.de

Abstract: The goal of this paper is to increase the computation speed of MapReduce jobs by reducing the accuracy of the result. Often, the timely processing is more important than the precision of the result. Hadoop has no built-in functionality for such an approximation technique, so the user has to implement sampling techniques manually. We introduce an automatic system for computing arithmetic approximations. The sampling is based on techniques from statistics and the extrapolation is done generically. This system is also extended by an incremental component which enables the reuse of already computed results to enlarge the sampling size. This can be used iteratively to further increase the sampling size and also the precision of the approximation. We present a transparent incremental sampling approach, so the developed components can be integrated in the Hadoop framework in a non-invasive manner.

1 Introduction

Over the last ten years, MapReduce [DG08] has become an often-used programming model for analyzing Big Data. Hadoop[1] is an open-source implementation of the MapReduce framework and supports executing jobs on large clusters. Different from traditional relational database systems, MapReduce focusses on the three characteristics ("The 3 Vs") of Big Data, namely volume, velocity and variety [BL12]. Thus, efficient computations on very large, fast changing and heterogeneous data are an important goal. One benefit of MapReduce is that it scales. So, it is well-suited for using the KIWI approach ("Kill It With Iron"): If a computation is too slow, one can simply upgrade to better hardware ("Scale Up") or add more machines to a cluster ("Scale Out").

In this paper, we focus on a third dimension additional to resources and time, namely computation accuracy. The dependencies of the dimensions can be depicted in a time-resources-accuracy triangle. It says that one cannot make Big-Data analyses in short time with few resources and perfect accuracy. The area of the triangle is constant. Thus, if one wants to be accurate and fast, more resources are needed (KIWI approach). If a hundred-percent accuracy is not mandatory, a job can run fast and without upgrading the hardware.

On the one hand, most work regarding Big-Data analysis, i.e. frameworks and algorithms are 100% precise. On the other hand, these approaches often give up the ACID properties

[1] http://hadoop.apache.org

and claim: Eventual consistency ("BASE") is enough. So, let us add this: For many computations, a ninety-percent accuracy is enough. One example: Who cares if the number of your friends' friends' friends in a social network is displayed as 1,000,000 instead of 1,100,000?

Some people extend the definition of Big Data by a fourth "V": *veracity* [Nor13]. This means, the data sources differ in their quality. Data may be inaccurate, outdated or just wrong. So, in many cases, Big-Data analyses are already inaccurate. When using sampling, the accuracy of the result decreases again, but the computation time improves. Sampling means, only a part of the data is analyzed and the results are extrapolated in the end.

Within this work, we extended the Marimba framework (see section 4.1) by a sampling component to execute existing Hadoop jobs with just a few changes in the user's jobs and no changes in the Hadoop framework. We use the so-called *Overwrite Installation* for making sampling jobs incremental. So, when increasing the size of samples, results from a former computation on a small sample size can be reused. For our framework, we use common sampling techniques which are presented in the following subsection.

In the following, some related work is presented. These are existing frameworks which also add sampling to MapReduce and techniques our approach is based on. In section 3, we explain commonly-used sampling techniques and present our approach for a Hadoop-based sampling framework. Section 4 focuses on making sampling incremental to increase the accuracy of a computation without starting from scratch. The sections 5 and 6 contain a performance evaluation and a conclusion.

2 Related Work

Hadoop RandomSampler There are different approaches for doing sampling in MapReduce. First of all, there is no built-in sampling in Hadoop. But for getting an idea of how the input data looks like, a *RandomSampler*[Whi09, p. 226] can be used. With that, Hadoop reads a given percentage of the input data (key-value pairs) which is stored on different machines and collects the keys of these. This is done before the Map phase and the result is a *key set*. This key set is a subset of all input keys and can be used to define key intervals for Map-output partitions. A *TotalOrderPartitioner* produces partitions with a better distribution than without sampling, so every Reducer has an equal amount of work. An example: Input keys are URLs of websites. Some of them start with "mail.", some with "mobile.", but more than 99% of all keys start with "www.". If the number of Reducers is three, a TotalOrderPartitioner would create three partitions, e.g. $[a-i]$, $[j-r]$ and $[s-z]$. But as there is no uniform distribution, 99% of the Map outputs will be sent to the third Reducer. With Random Sampling, the partitions are of the same size, e.g. $[a-www.k]$, $[www.k-www.q]$ and $[www.r-z]$. This kind of sampling can be used when Reducers need their keys in the right order. As this is not needed for most MapReduce jobs, a simple *HashPartitioner* ensures a good distribution by performing modulo hashing. Nevertheless, different from our approach, this sampling technique fails to increase computation performance.

EARL The *Early Accurate Result Library* (EARL) [LZZ12] is a modification of the Hadoop framework to run MapReduce jobs on a subset of the data. This subset is randomly created in a *Sampling Stage*, the first of three phases. In the second phase, *resamples* are created which are subsets of the given sample. Each of these resamples is used as input for an individual MapReduce computation. The outputs of these are compared to each other to determine the accuracy. This is done in the third phase, called *Accuracy Estimation*. EARL changes Hadoop very deeply. It forces MapReduce jobs not to stop after they are finished and to accept more input data by increasing the input-sample size. This is done until a desired accuracy is reached.

EARL provides two strategies for reading a subset of the input, namely *Pre-Map sampling* and *Post-Map sampling*. As their names imply, the first strategy is to skip most of the input tuples and only perform the Map function on p% of the input. In the second strategy, the Map function is computed on the full dataset, but if it wants to emit data, this is only done with a probability of p%. So, most of the Map output is discarded. Pre-Map sampling is faster than Post-Map sampling because the Map function is called less often, but it leads to a lower accuracy because it either processes a segment completely or not at all. In both Pre-Map and Post-Map sampling, the Reducers only get a subset of the complete intermediate-key/value pairs. In the second approach, they are more random. EARL does not extrapolate the Reduce outputs automatically. Instead, the user has to provide a *Correct* function. So, if for example a $p\%$ sample of a large text is read to count the occurrences of terms in the text, the Correct function has to multiply the count values in the end by $100/p$ to estimate the real quantities. In our framework, one does not need to provide a Correct function and our framework does not modify the Hadoop core.

Sampling for Hadoop Other works focuses only on sampling the input. Our approach also extrapolates the results. In [GC12], Grover and Carey tested how to increase the performance of Hadoop when doing *predicate sampling*, i.e. finding n items in a given dataset which fulfill a predicate. The algorithm stops when enough items are found. The randomness of the sampling set is not important for this algorithm. So, the results can not be used to estimate properties of the full dataset well.

Apache's Pig provides a *sample* operator[2] for shrinking a dataset. The following Pig script will produce a 10% sample: `a = LOAD 'data.csv'; s = SAMPLE a 0.1`

Sampling in Relational DBMS In [OR90], Olken and Rotem suggest sampling as a DBMS operator. This leads to changes in the query language, optimizers and access paths. It is described how B+ trees and hashing algorithms have to be modified for accessing only a specific percentage of the data. The objectives are estimating the results of aggregate functions with the usage of sampling techniques. In [PBJC11], these techniques are built into MapReduce. However, only simple aggregate functions after a grouping are supported. In [PJ05], bootstrapping is used for relational databases and again, the approximation method which is used there only supports simple aggregates. In [Fis11], an interactive data-visualization tool is presented that incrementally delivers more accurate

[2] https://pig.apache.org/docs/r0.11.1/basic.html#sample

results to a given query. Instead of returning an inaccurate result value to the user, the result is presented as a range of possible values. The tool can be used to produce many kinds of histograms based on GROUP BY queries.

3 Sampling with Hadoop

Sampling is used to draw a subset of a population. Afterwards the characteristics of the population are estimated based on the characteristics of the subset. The goal is to get a representative subset in order to get more accurate extrapolations. There are different techniques for choosing a subset [Sud76]. In the following, three of them are described (see Figure 1).

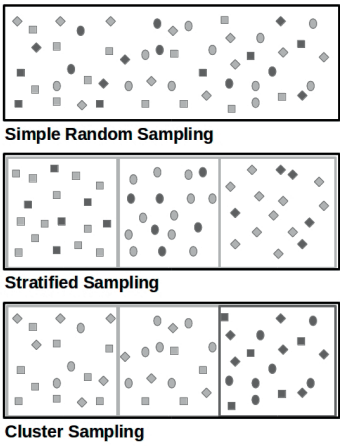


Figure 1: Three Sampling Techniques

Simple Random Sampling When using *Simple Random Sampling*, all elements are chosen with the same probability of $p\%$. According to the *Law of Large Numbers*, the subset of a population of size n will have the size $n \cdot p/100$.

Stratified Sampling In the *Stratified-Sampling* approach, the population is divided into sub-populations. On each of these, sub-population sampling is done. To improve the accuracy, the dividing characteristics should correlate with the focused characteristic. Additionally, the sub-populations should be homogeneous.

Cluster Sampling This approach also divides the population in subpopulations (clusters), but this division is done *naturally*, based on for example location, etc. Afterwards one or more clusters are randomly chosen and are completely used as the subset for the extrapolation. There is the risk, that the extrapolation might be biased because of homogeneous clusters. It is possible to reduce that risk by using a *Simple Random Sampling* on the chosen clusters.

Our goal is to leave the MapReduce framework intact. So we first show how to change an existing Hadoop job to improve the performance at an expense of accuracy. Later in this section, we present a framework which can be used on top of Hadoop to execute existing Hadoop jobs with the usage of sampling [Sch14]. With that, the user should not have to change her Map and Reduce functions depending on whether the whole input or only a subset is analyzed.

3.1 Restrictions

There are three kinds of MapReduce jobs which must be treated differently when enabling sampling techniques on them.

Sampling only Top-k queries, computations of MIN, MAX or AVG, and the determination of a set of items with certain properties are some examples for jobs with no need for an extrapolation. The result of a computation on a subset of the input can be used as a final result. Inaccuracies arise through missing hits: Maybe another skipped value is higher than the computed maximum and maybe some skipped items fulfill the search criteria as well. The average function does not need to be extrapolated because it is a division of a sum and a count. The extrapolation would be canceled out.

Sampling and Extrapolation Counting and computing sums are popular tasks in MapReduce jobs. If these operations are performed on a subset, the result has to be multiplied by $100/p$ where p is the sampling percentage. Other tasks which produce complex objects or compute the final results by a multiplication need different extrapolation functions. For the latter one the result r has to be corrected like this: $r^{100/p}$.

Sampling not supported Iterative algorithms, graph transformations and more do not support sampling because the output computed on a subset is useless. They always have to be executed on the full input.
In our work, we focused on the *Sampling and Extrapolation* class of jobs. It stands to reason that *Sampling only* jobs are supported too. Here, the extrapolation can be seen as the identity function.

3.2 Subset selection

As it is done in EARL (see section 2), a user can implement a Pre-Map or a Post-Map sampling by either changing the *RecordReader* of the used *InputFormat* to skip $(100-p)\%$ of the items. Or one adds some lines to the user-defined Map function to skip input or output key-value pairs with a probability of $p\%$.

Our goal is an automated subset selection. Therefore, we propose a *SamplingInputFormat* which delegates all function calls to the actual user-defined InputFormat and is able to skip items in the input splits (*item sampling* / for texts: *line sampling*), or to skip splits as a whole (*split sampling*). A split is a chunk of items which are stored on one machine and processed by one Mapper node. When using text files which are stored in the Hadoop Distributed File System (HDFS), a split usually is a 64 MB large text snippet. The lines in this snippet are called *items* and are the input for one Map call.

A selection based on the set of items corresponds to the *Simple Random Sampling* technique. Thereby, a potential uniform coverage is given, since the selection is done on the smallest usable unit. The disadvantage of this technique is that the RecordReader has to read every item and discard most of them.

A less time-consuming approach is a selection on the set of splits. This approach corresponds to *Cluster Sampling* where the splits are the naturally-grouped subsets based on locality. This approach discards whole splits, which is significantly less computation than in the previous approach. If the data is not uniformly distributed, the split approach could lead to biased results.

A user can balance the pros and cons of both approaches by combining them. Our frame-

work provides the configuration of two percentage values for both item and split selection, so one can perform a 10% sampling by skipping every second split and in each split doing a 20% item sampling.

Jobs of the class *Sampling only* don't need any further modifications or considerations. Item sampling, split sampling and the combination of both are all Pre-Map approaches. The user-defined Map function is called for the subset of items the *SamplingInputFormat* provides. The Map and Reduce functions can stay unchanged and no extrapolation is needed.

3.3 Extrapolation

For jobs of the class *Sampling and Extrapolation*, the final output has to be corrected to estimate the result of a computation on the whole data. Approaches presented in section 2 rely on a user-defined Correct function. This function is executed *Post-Reduce*. We want to introduce a generic *Pre-Reduce* extrapolation method. The benefit of that is that Map and Reduce can be left unchanged and that no Correct function is needed.

Example (Post-Reduce / Pre-Reduce Extrapolation) A MapReduce job is used to compute the number of website visits per day by analyzing log files. 5% of the input is sampled and passed through the Map function. Using *Post-Reduce Extrapolation*, the Reduce function sums up all visit counts for one day. After that, a Correct function has to multiply the numbers by 20. With *Pre-Reduce Extrapolation*, before calling the Reduce function, each intermediate key-value pair is copied 19 times so it reaches the Reducer twenty times. The output in both approaches is the same.

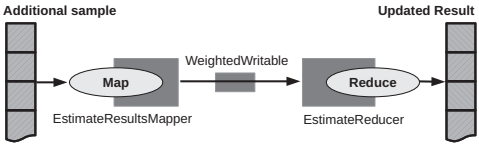


Figure 2: The extrapolation based on the old result and a new sample.

At a first glance, the Pre-Reduce approach sounds expensive because the amount of data shipped to the Reducers increases drastically. We solved this by copying the data just virtually by encapsulating the user-defined Map-output type and the Reduce function (see Figure 2). A *WeightedWritable* object contains one Map-output value and a weight value (in the example above it would be 20). The user-defined Reducer is replaced by an *EstimateReducer* that delegates each function call to the actual Reducer but with replacing the Iterable object that contains all values by a *MultiIterable*. This is programmed to return each value n times when calling the `next()` function.

With this approach, a problem arises for a non-integer n since an element cannot be reused for example 0.6 times. A simple solution would be to round the value, but then the accuracy of the result is too low. For example, when sampling a 62.5% subset, $n = 1.6$. Rounding would lead to $n = 2$, which would lead to a simulated dataset of 125% of the original dataset size. We solved this problem by iterating over each element $\lfloor n \rfloor$ times and with a probability $n - \lfloor n \rfloor$ once more. So, if $n = 2.6$, each element is either processed two or three times. The probabilities for that are 40% respectively 60%. As it is shown later, this approach leads to good results.

4 Incremental Sampling

The previous section showed how to extend Hadoop with a generic sampling approach. But it is hard to determine which sampling percentage to choose to deliver a sufficient accuracy. One is not able to tell the accuracy of a computation without comparing the result to the result of the full computation. EARL (see section 2) estimates the accuracy by sampling multiple times and comparing the results. We use an approach that is often used in iterative graph algorithms like PageRank [PBMW99]. If the change of the results towards a former iteration is below a threshold, no further iteration is needed. This means for sampling, one starts with a small sample size and increases this size step by step until the results converge. To increase the precision of an already computed estimation, it is not necessary to discard the former result. Instead, it is sufficient to make a new computation and combine its results with the former one. Our incremental approach is based on the Marimba framework.

4.1 Marimba Framework

A Hadoop-based framework, called Marimba [SJHD14] can be used to create self-maintainable MapReduce jobs. Based on the two strategies *Overwrite* and *Increment Installation* presented in [JPYD11], Marimba executes a MapReduce computation only on the changes in the base data since the former computation and aggregates the results to the former results. If this aggregate operation is a simple plus operation and if the output format supports in-place additions, the *Increment Installation* can be used. HBase has a support for this strategy, so if for example a Reduce output key/value pair is `('high', 5)`, the current value for the key `high` is read, it is increased by five and written back. As this read operation is not a sequential but a random read, the Increment Installation is very expensive if there are many changes in the input since the former computation. Therefore, the *Overwrite Installation* can be used. Here, the full former result is read and aggregated to the Map output records of the changed data. In the end, the final result overwrites the former one. A benefit of Marimba is that the user can simply execute a normal Hadoop job with only making a few changes. First, the Map-output values have to form an Abelian group. This can be reached by replacing the Hadoop type *LongWritable* with the invertible Marimba type *LongAbelian*. Second, the user has to define a *Deserializer* to reuse the former result.

4.2 Reusing Former Sampling Results

A weighting of the inputs is necessary since every input can have a different size. One example: After a one-percent sampling, another percent is sampled and the results are combined. As both results are too different, another percent should be sampled. Now, the former result has to be double-weighted and the new data single-weighted. The end result is then based on three percent of the input data. Executing multiple incremental one-percent jobs is much faster than classical non-incremental jobs. As a drawback, it leads

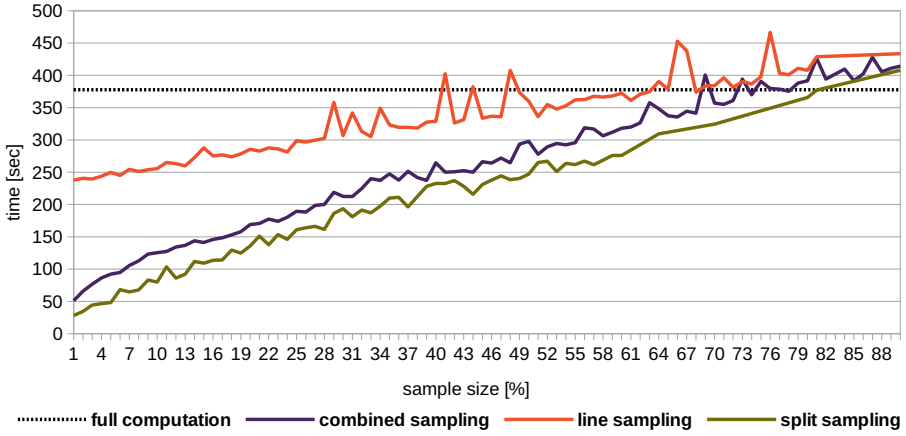


Figure 3: The three sampling approaches in comparison.

to a shade lower accuracy because of overlaps in the input samples. Incremental sampling corresponds to *sampling with replacement*, classical approaches to *sampling without replacement*. Note that the maximum number of replacements is limited by the number of iterations.

An incremental sampling job needs to read two inputs: a former result and an additional sample. To reuse the already computed result, another Map function is needed to reverse the extrapolation from the former iteration. This function converts Reduce output key/value pairs to Reduce input key/value pairs. In many MapReduce jobs, the input and output type are equal. Then an identity mapping is sufficient. As mentioned in the previous chapter, the *WeightedWritable* class is used as intermediate-value class. The weights for the reused results depends on the former sample size.

Our incremental-sampling framework uses the Overwrite Installation approach [JPYD11]. The incremental computation is divided into two steps. The first step is a normal MapReduce sampling job for a new subset of the input data. The result of this job together with a former result are the inputs for the second job. This second job is the incremental part which computes the updated result. For this job, the user needs to develop a Map function which reads the data and the old results and converts these into key/value pairs. Also, a Reduce function is needed to write the new result. Weighting is automatically done after the user-defined Map function through the intermediate-value class *WeightedWritable* which contains the value together with the weighting factor. A special Reduce function is needed to process this intermediate results. This function computes the final Reduce outputs based on the input values and the weighting factor.

5 Evaluation

We tested our sampling framework on a five-node Hadoop cluster. Each node consists of a Quadcore 2.53 GHz CPU and 4 GB of main memory. They are connected via Gigabit Ethernet.

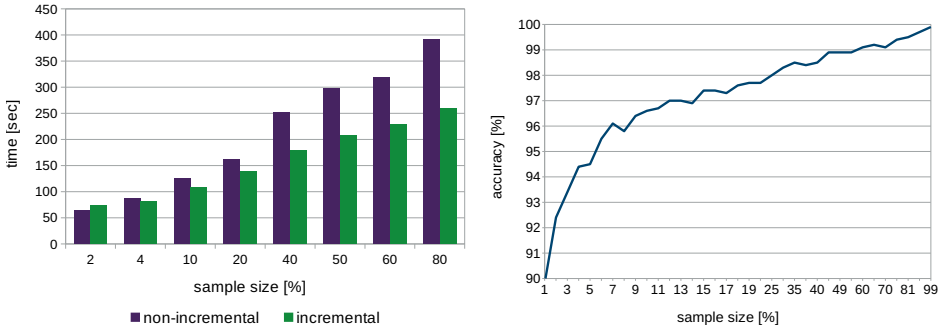


Figure 4: (a) Non-incremental vs. incremental sampling (b) Accuracy of a computation

Figure 3 shows the runtimes of the three sampling approaches. All tests are based on a word-count job on 3200 English texts of a size of 1.2 GB. As every split is read in the line-based sampling, it is slower than split-based sampling. The combination of both approaches[3] lies in between. When the sampling size is more than 50%, a full computation is faster than using sampling because the amount of intermediate data includes a few overhead and has to be shipped over the network.

In Figure 4a, a $(p/2)\%$ sample was doubled to $p\%$. When using incremental sampling, the old result can be reused which leads to a better performance than non-incremental sampling for p greater than two percent. We compared the results of the word-count sampling job with the actual count values and observed an accuracy of more than 90% for the words that occurred more than 5000 times (see Figure 4b).

Our assumption and the time-resources-accuracy triangle in the beginning of this paper were confirmed: For a sampling size of 10%, we saw that our incremental sampling approach is three times faster than a full computation at the expense of 3% inaccuracy.

6 Conclusion

The goal of the presented framework is to provide user-friendly sampling and extrapolation components for Hadoop which also enable an incremental sampling approach. These components can be used for sampling of arithmetic jobs. We made experiments on those jobs and we found out that there is a large speed-up when using sampling. This speed-up is even higher when using incremental sampling where former results can be reused to increase the sample size. Different from other sampling approaches, our framework does not need to modify the Hadoop framework. A user can simply enable input-data sampling and an automatic extrapolation of the results for existing MapReduce jobs with only a few modifications.

In future work, our sampling approaches can be used in an interactive data-visualization tool. Like in [Fis11], a user can see the results of a query in a chart. When using a small sample size, the chart can be displayed very fast. With the usage of incremental sampling,

[3]For a sampling size $s \in (0, 1]$, $\sqrt{s}$ of the splits are read, and within a split, $\sqrt{s}$ of the lines. For example, a 25% sampling ($s = 0.25$) is divided into 50% of splits and 50% of lines.

it can become more accurate over time. Currently, our framework does not give accuracy guarantees. We are currently working on a new version which displays the accuracy to the user.

References

[BL12] Mark A Beyer and Douglas Laney. The importance of'big data': a definition. *Stamford, CT: Gartner*, 2012.

[DG08] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[Fis11] Danyel Fisher. Incremental, approximate database queries and uncertainty for exploratory visualization. In *Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on*, pages 73–80. IEEE, 2011.

[GC12] Raman Grover and Michael J Carey. Extending map-reduce for efficient predicate-based sampling. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 486–497. IEEE, 2012.

[JPYD11] Thomas Jörg, Roya Parvizi, Hu Yong, and Stefan Dessloch. Incremental recomputations in MapReduce. In *Proceedings of the third international workshop on Cloud data management*, CloudDB '11, pages 7–14, New York, NY, USA, 2011. ACM.

[LZZ12] Nikolay Laptev, Kai Zeng, and Carlo Zaniolo. Early Accurate Results for Advanced Analytics on MapReduce. *CoRR*, abs/1207.0142, 2012.

[Nor13] K Normandeau. Beyond Volume, Variety and Velocity is the Issue of Big Data Veracity. *Inside Big Data*, 2013.

[OR90] Frank Olken and Doron Rotem. Random sampling from database files: A survey. In *Statistical and Scientific Database Management*, pages 92–111. Springer, 1990.

[PBJC11] Niketan Pansare, Vinayak R Borkar, Chris Jermaine, and Tyson Condie. Online aggregation for large mapreduce jobs. *Proc. VLDB Endow*, 4(11):1135–1145, 2011.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. 1999.

[PJ05] Abhijit Pol and Christopher Jermaine. Relational confidence bounds are easy with the bootstrap. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 587–598. ACM, 2005.

[Sch14] Marc Schäfer. Samping mit inkrementellem MapReduce. Master's thesis, Technische Universität Kaiserslautern, 2014.

[SJHD14] Johannes Schildgen, Thomas Jörg, Manuel Hoffmann, and Stefan Deßloch. Marimba: A Framework for Making MapReduce Jobs Incremental. *IEEE International Congress on Big Data*, 2014.

[Sud76] Seymour Sudman. *Applied sampling*. Academic Press New York, 1976.

[Whi09] Tom White. *Hadoop: the definitive guide: the definitive guide*. " O'Reilly Media, Inc.", 2009.

METIS in PArADISE
Provenance Management bei der Auswertung von Sensordatenmengen für die Entwicklung von Assistenzsystemen

Position Statement
Andreas Heuer, Universität Rostock, Institut für Informatik

andreas.heuer@uni-rostock.de

1 Einleitung

Wissenschaftliche Experimente, die durch Messungen oder ständige Beobachtungen laufend eine Vielzahl von Daten erheben, müssen durch effiziente Analyseverfahren auf diesen Mess- oder Sensordaten unterstützt werden. Die Mess- und Sensordaten sind so zu verwalten, dass sie im Sinne des Provenance Management rückverfolgbar werden. An der Universität Rostock soll ein langfristiges Forschungsvorhaben im Bereich der Informatik und Elektrotechnik etabliert werden, in dem wissenschaftliche Experimente in der Informatik, der Zellbiologie und der Medizin (neurodegenerative Erkrankungen) auf eine solche Weise unterstützt werden.

Im Bereich der Informatik ist das experimentelle Anwendungsgebiet das der Erforschung und systematischen Entwicklung von Assistenzsystemen. Da in Assistenzsystemen unterstützte Personen durch eine Vielzahl von Sensoren beobachtet werden, müssen auch Privatheitsaspekte bereits während der Phase der Modellbildung berücksichtigt werden, um diese bei der konkreten Konstruktion des Assistenzsystems automatisch in den Systementwurf zu integrieren. Somit gibt es für die Datenbankforscher unter anderem folgende Teilprobleme, die zu lösen sind und die in zwei langfristigen Projektgruppen des Datenbanklehrstuhls zusammengefasst sind:

PArADISE (Privacy AwaRe Assistive Distributed Information System Environment): In dieser Projektgruppe werden Techniken zur Auswertung von großen Mengen von Sensordaten entwickelt, die definierte Privatheitsansprüche der späteren Nutzer per Systemkonstruktion erfüllen.

METIS (Management, Evolution, Transformation und Integration von Schemata): In dieser Projektgruppe geht es unter anderem um Integrationsverfahren für heterogene Datenbanken, die per se bereits eine Rolle bei der Zusammenführung der unterschiedlichen Datenquellen des Assistenzsystems spielen. Als Seiteneffekt werden die für die Datenintegration entwickelten Techniken der *Global-as-local-view-extension* (GaLVE) auch für

die nötigen inversen Abbildungen zum Provenance Management bei der Verarbeitung der Sensordaten und Ableitung von (Situations-, Aktivitäts-, Intentions-)Modellen benötigt.

Wissenschaftliche Experimente außerhalb der Informatik, etwa zur Auswertung von Sensordaten bei Patienten mit neurodegenerativen Erkrankungen [DEW+12] liegen nicht im Fokus dieses Beitrags.

Im Folgenden werden wir im Abschnitt 2 kurz die Architektur von Assistenzsystemen einführen, wobei wir den Schwerpunkt auf die Phase der Situations-, Aktivitäts- und Intentionserkennung legen. Danach werden wir die Erforschung und Entwicklung von Assistenzsystemen als ein wissenschaftliches Experiment ansehen, in dem Forscher eine große Anzahl von Sensordaten auswerten müssen und aus ihnen Situations-, Aktivitäts- und Intentionsmodelle entwickeln (Abschnitt 3). Zwei der Grundlagenforschungsthemen für die Datenbankforscher, Provenance Management und die Integration von Privatheitsanforderungen, werden wir in Abschnitt 4 einführen.

2 Assistenzsysteme

Ähnlich einem menschlichen Assistenten soll ein Assistenzsystem mich unterstützen, im Hintergrund arbeiten (ambient), mich nicht stören, zum richtigen Zeitpunkt eingreifen und Hilfe anbieten (diese in üblichen Fällen auf optischem oder akustischem Wege), vertrauenswürdig und diskret sein und sich bei Bedarf abschalten lassen.

Um seine Assistenzaufgaben zu erfüllen, besteht ein Assistenzsystem üblicherweise aus fünf Schichten (siehe Abbildung 1). Dabei deutet die Pyramidenform an, dass in der untersten Schicht dauernd viele Daten (etwa von Sensoren) erzeugt werden, in der obersten Schicht aber nur im Bedarfsfall (also eher selten) ein akustischer oder optischer Hinweis, also eine geringe Datenmenge, ausgegeben wird.

Sensoren in der Umgebung der Person sollen Situation und Tätigkeit der Person erfassen, um ihr assistieren zu können. **Ortungskomponenten** sollen die genaue Position der Person bestimmen, etwa zur Detektion dementer Patienten mit Weglauftendenzen. Sensoren und Ortungskomponenten befinden sich in der Umgebung, in benutzten Geräten oder am Körper der Person (Armband, Brille, ...).

Damit ein Assistenzsystem seine Aufgabe erfüllen kann, müssen verschiedene (heterogene) Geräte in der Umgebung der Person **vernetzt** und zur Erreichung des Assistenzziels **spontan gekoppelt** werden.

Sensordaten müssen gefiltert, erfasst, ausgewertet, verdichtet und teilweise langfristig verwaltet werden. Aufgrund der extrem großen Datenmenge (Big Data) muss die **Verarbeitung verteilt** erfolgen: teilweise eine Filterung und Verdichtung schon im Sensor, im nächsterreichbaren Prozessor (etwa im Fernseher oder im Smart Meter in der Wohnung) und im Notfall über das Internet in der Cloud. Neben Daten des Assistenzsystems müssen auch fremde Daten etwa über das Internet berücksichtigt werden.

Nach Auswertung der Sensordaten erfolgt die **Situations-, Handlungs- und Intentionserkennung** (siehe den folgenden Abschnitt 3 für die Entwicklung diesbezüglicher Modelle).

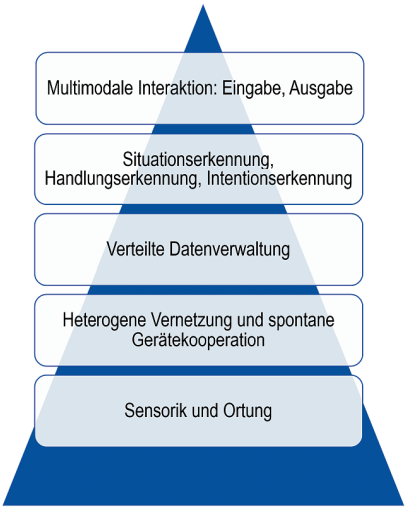


Abbildung 1: Pyramidenarchitektur von Assistenzsystemen

3 Big Data Analytics für die Entwicklung von Assistenzsystemen

Aufgrund der Sensor- und Ortungsdaten sowie der weiteren über das Internet erhältlichen Daten muss das Assistenzsystem eine Situations- und Handlungserkennung vornehmen sowie eine Handlungsvorhersage (Intentionserkennung), um proaktiv eingreifen zu können.

Die Situation ist dabei die aktuelle Umgebungsinformation, die Handlung das, was die Person, der assistiert wird, gerade durchführt. Die Intentionserkennung oder Handlungsvorhersage muss voraussagen, was die die Person in Kürze tun wird.

Auch die Handlungs- und Intentionserkennung ist ein aktueller Forschungsgegenstand der Informatiker an der Universität Rostock [KNY+14]. Dabei erheben die Forscher in langen Versuchsreihen eine extrem hohe Anzahl von Sensordaten, aus denen sie mit diversen Analyseverfahren die entsprechenden Modelle ableiten. Diese Analyseverfahren sind — wenn sie ohne Datenbankunterstützung auf Dateisystemen mit Analysewerkzeugen wie R ausgeführt werden — mehrwöchige Prozesse. Ziel der Forscher ist neben der Modellbildungen für Handlung und Intention die Erkenntnis, wie die große Anzahl von Sensoren im Versuch für den praktischen, späteren Einsatz des Assistenzsystems drastisch eingeschränkt werden kann, ohne die Vorhersagequalität zu mindern. Für die Ableitung dieser Informationen müssen unter anderem alle Analysefunktionen invertiert werden, um die für die Modellbildung entscheidenden Anteile der Originaldaten zu finden. Letzteres ist auch ein Problem im Provenance Management, das die Experimentverläufe mit den Ergebnisableitungen begleiten soll. Eine Einschränkung sowohl der Anzahl der Sensoren als auch der Menge und Granularität der erfassten Daten ist auch aus einem anderen Grund wichtig: sie kann die Privatheitsanforderungen der Nutzer des Assistenzsystems realisieren helfen.

4 Nötige Grundlagenforschung: Privatheit und Provenance

Die Entwickler der Analysewerkzeuge müssen ihr Assistenzziel und die notwendigen Sensordaten zur Erreichung des Ziels und zur grundlegenden Situations-, Handlungs- und Intentionserkennung formulieren. Diese Zielformulierung wird dann in Anfragen auf Datenbanken transformiert. Weiterhin können die **Privatheitsansprüche** des Nutzers vordefiniert oder von jedem Nutzer selbst individuell verschärft werden. Auch diese Privatheitsansprüche werden in Anfragen (Sichten) auf Datenbanken umgesetzt. Durch Abgleich des Informationsbedarfs des Assistenzsystems und der Privatheitsansprüche des Nutzers kann dann die Datenbankkomponente des Assistenzsystems entscheiden, wie die Menge an Sensordaten selektiert, reduziert, komprimiert oder aggregiert werden muss, um beiden Parteien im System gerecht zu werden[Gru14].

Ein entscheidendes Kriterium für die Vertrauenswürdigkeit eines Assistenzsystems ist noch die Frage, wie nah am Sensor die Daten bereits reduziert und verdichtet werden können: Wenn der Sensor so intelligent ist, dass er bestimmte Filtermechanismen von Datenbanksystemen beherrscht, so kann dieser bereits eine Vorfilterung vornehmen. Nur die für das Assistenzziel unabdingbaren Daten, die die Privatheit des Nutzers nicht verletzen, können dann im Rahmen des Cloud Data Management des Anbieters der Assistenzfunktionalität entfernt und verteilt gespeichert werden.

Ein Teil der Datenbankforschung im Gebiet **Provenance** konzentriert sich auf einzelne Modelle, Operatoren und Annotationen für die Rekonstruktion der Rohdaten [CAB+14]. Eine durchgängige und automatisierte Vorgehensweise fehlt jedoch. Daneben wurden Provenance-Eigenschaften charakterisiert und formalisiert [BKT01], meist für das Tracing der Daten vorwärts [GKT07] und nicht für die automatische Rekonstruktion rückwärts. Für Letzteres werden Charakterisierungen inverser Analyseoperatoren und inverser Schemaabbildungen benötigt. Inverse Schemaabbildungen und ihre Eigenschaften werden zwar untersucht [Fag07], derzeit aber noch nicht im Bereich Provenance angewandt, sondern eher bei der Schema-Evolution [CMDZ10].

In [FHLM96] wurde die Idee der **inversen Schemaabbildungen** bereits für die Integration heterogener Datenbanken eingeführt. Diese Idee wurde in [SBLH14] auf neuere Entwicklungen in der Theorie inverser Schemaabbildungen angepasst. Dabei wurden die klassischen Schemaabbildungen verallgemeinert auf den Fall, dass nicht alle relevanten Daten im (relationalen) Schema repräsentiert werden können, sondern zusätzlich auf Instanzebene erfasst werden müssen. Diese zusätzlichen Annotationen helfen bei der inversen Abbildung verdichteter Daten.

Im Gegensatz zur traditionellen Datenintegration sollen Anfragen im integrierten System bei diesem Ansatz nicht an das globale Schema gestellt werden, sondern weiterhin an die lokalen Schemata, wobei diese um Daten aus anderen Quellen erweitert werden. Dieser Ansatz wird als *Global-as-local-view-extension* (GaLVE) bezeichnet.

Inverse Schema-Instanz-Abbildungen werden nun auch für die Invertierung, also Rückverfolgung, von Analyseprozessen, also allgemeineren Datenbankanfragen, benötigt. Bisher wurden allerdings für die Datenbankintegration nur einfache Anfragen, bestehend aus Selektion, Projektion und Verbund, berücksichtigt. Analyseprozesse erfordern eine Erwei-

terung der bestehenden Technik auf statistische Funktionen (skalare Funktionen, Aggregatfunktionen, OLAP) und allgemeine Workflows. Hier muss ermittelt werden, welche Zusatzinformationen zur Gewährleistung der Rückverfolgbarkeit erfasst werden müssen.

Da in diesem seitenbeschränkten Beitrag leider viele Aspekte und Literaturhinweise nicht aufgenommen werden konnten, findet sich eine Langfassung dieses Position Papers unter `www.andreas-heuer.de/files/metis-in-paradise-long.pdf`

Literatur

[BKT01] Peter Buneman, Sanjeev Khanna und Wang Chiew Tan. Why and Where: A Characterization of Data Provenance. In Jan Van den Bussche und Victor Vianu, Hrsg., *ICDT*, Jgg. 1973 of *Lecture Notes in Computer Science*, Seiten 316–330. Springer, 2001.

[CAB+14] Lucian Carata, Sherif Akoush, Nikilesh Balakrishnan, Thomas Bytheway, Ripduman Sohan, Margo Seltzer und Andy Hopper. A Primer on Provenance. *Commun. ACM*, 57(5):52–60, Mai 2014.

[CMDZ10] Carlo Curino, Hyun Jin Moon, Alin Deutsch und Carlo Zaniolo. Update Rewriting and Integrity Constraint Maintenance in a Schema Evolution Support System: PRISM++. *PVLDB*, 4(2):117–128, 2010.

[DEW+12] Martin Dyrba, Michael Ewers, Martin Wegrzyn, Ingo Kilimann, Claudia Plant, Annahita Oswald, Thomas Meindl, Michela Pievani, Arun L. W. Bokde, Andreas Fellgiebel, Massimo Filippi, Harald Hampel, Stefan Klöppel, Karlheinz Hauenstein, Thomas Kirste und Stefan J. Teipel. Combining DTI and MRI for the Automated Detection of Alzheimer's Disease Using a Large European Multicenter Dataset. In Pew-Thian Yap, Tianming Liu, Dinggang Shen, Carl-Fredrik Westin und Li Shen, Hrsg., *Multimodal Brain Image Analysis - Second International Workshop, MBIA 2012, Held in Conjunction with MICCAI 2012, Nice, France, October 1-5, 2012. Proceedings*, Jgg. 7509 of *Lecture Notes in Computer Science*, Seiten 18–28. Springer, 2012.

[Fag07] Ronald Fagin. Inverting schema mappings. *ACM Trans. Database Syst.*, 32(4), 2007.

[FHLM96] Guntram Flach, Andreas Heuer, Uwe Langer und Holger Meyer. Transparente Anfragen in föderativen Datenbanksystemen. In *Proceedings zum Workshop Föderierte Datenbanken*, Seiten 45–49, 1996.

[GKT07] Todd J. Green, Gregory Karvounarakis und Val Tannen. Provenance semirings. In Leonid Libkin, Hrsg., *PODS*, Seiten 31–40. ACM, 2007.

[Gru14] Hannes Grunert. Distributed Denial of Privacy. In Erhard Plödereder, Lars Grunske, Eric Schneider und Dominik Ull, Hrsg., *44. Jahrestagung der Gesellschaft für Informatik, Informatik 2014, Big Data - Komplexität meistern, 22.-26. September 2014 in Stuttgart, Deutschland*, Jgg. 232 of *LNI*, Seiten 2299–2304. GI, 2014.

[KNY+14] Frank Krüger, Martin Nyolt, Kristina Yordanova, Albert Hein und Thomas Kirste. Computational State Space Models for Activity and Intention Recognition. A Feasibility Study. *PLOS ONE*, November 2014. 9(11): e109381. doi:10.1371/journal.pone.0109381.

[SBLH14] Georgi Straube, Ilvio Bruder, Dortje Löper und Andreas Heuer. Data Integration in a Clinical Environment Using the Global-as-Local-View-Extension Technique. In *HIS*, Jgg. 8423 of *Lecture Notes in Computer Science*, Seiten 148–159. Springer, 2014.

Extracting reproducible simulation studies from model repositories using the CombineArchive Toolkit

Martin Scharm, Dagmar Waltemath

Department of Systems Biology and Bioinformatics
University of Rostock
D-18051 Rostock
martin.scharm@uni-rostock.de
dagmar.waltemath@uni-rostock.de

Abstract:
The COMBINE archive is a digital container format for files related to a virtual experiment in computational biology. It eases the management of numerous files related to a simulation study, fosters collaboration, and ultimately enables the exchange of reproducible research results. The CombineArchive Toolkit is a software for creating, exploring, modifying, and sharing COMBINE archives. Open model repositories such as BioModels Database are a valuable resource of models and associated simulation descriptions. However, so far no tool exists to export COMBINE archives for a given simulation study from such databases. Here we demonstrate how the CombineArchive Toolkit can be used to extract reproducible simulation studies from model repositories. We use the example of Masymos, a graph database with a sophisticated link concept to connect model-related files on the storage layer.

1 Reusability of simulation studies in systems biology

A study by the pharmaceutical company Bayer in 2011 showed that only 64% of published data in academic journals were replicable [PSA11]. This and similar observations led to initiatives and projects that aim at improved result reproducibility. For example, the Reproducibility Initiative[1] is a collaboration between Science Exchange, PLOS, figshare and Mendeley, and it identifies and rewards high quality reproducible research via independent validation of key experimental results. One example for a European project is FAIRDOM[2]. It is a collaboration of data management groups who will develop the necessary toolset to extend existing network services to the wider European Systems Biology community. Furthermore, funders require explicit data management strategies in grant applications. Trans-project data management systems such as the SEEK platform [WO+11] have become an integral part of the scientific landscape. Finally, various projects work towards reproducible experiments by re-thinking the publication process. For example, Research Objects [BDR+10] support the publication of data, code and other resources

[1] http://reproducibilityinitiative.org/
[2] http://fair-dom.org/

alongside the PDF paper. A call for Virtual Experiments [CVW14] highlights the actual benefits of generic simulation setups for the reusability of full behavioural repertoires of computational models.

Virtual experiments in systems biology projects comprise of a variety of files in heterogeneous formats [HWW14]: the encoding of the model, the associated simulation setups, the semantic description of the underlying biological and mathematical concepts, the graphical description of the modelled processes, reference publications, supporting figures, data tables for model parametrisation, result data sets, experimental data sets used to verify the simulation results etc. One achievement of the Computational Modeling in Biology Network (COMBINE, [WB+14]) is the implementation of guidelines and data formats for the standardised representation of this data. While the various types of model-related data remain heterogeneous, the data of each single type is well described. For example, models can be encoded in the Systems Biology Markup Language (SBML, [HF+03]), an XML format that is supported by over 100 software tools today [HB+11]. The semantics of a model can be specified through links into bio-ontologies [CJ+11], and in fact most publicly available SBML models are also curated and annotated. A format for the visual representation of a model is the Systems Biology Graphical Notation (SBGN, [LNH+09]). It defines standard glyphs for common biological objects, their properties and their interactions. Simulation setups can be described in the Simulation Experiment Description Markup Language (SED-ML, [WA+11]), enabling their immediate execution in all software tools that read the format.

In summary, the current situation is such that standards exist to encode model-related data, the data is publicly available in model repositories, it can be linked, and it can be queried. In this paper, we show how the retrieved data can be bundled as COMBINE archives. Our prototype implementation showcases a workflow that combines two existing software tools: A graph database for linked storage of model-related data (Masymos) and a web-based tool to generate COMBINE archives (CAT). Using M2CAT (Masymos to CAT) a model can easily be retrieved together with all other files that are necessary to understand the model and to reproduce the original results.

2 Exporting COMBINE archives from Masymos

The presented workflow integrates two of our previously developed software tools: Masymos, a graph database to store model files, simulation descriptions and other model-related data; [HWW14] and the CombineArchiveWeb tool [SW+14], an online application to manage COMBINE archives.

Masymos: Storing model-related data. Model management includes tasks that aim to optimise model storage, search, retrieval, provenance, visualisation, and comparison. The increased awareness of data management needs among researchers and funders requires novel ways of ensuring reproducibility, distribution, and management of modelling results. Masymos is a graph database of models and model-related data in computational biology [HWW14]. It is a Neo4J database that contains annotated models in SBML and

CellML formats, simulation descriptions in SED-ML format, and links to reference publications and bio-ontologies. Masymos implements a sophisticated link concept to connect the heterogeneous types of model-related data on the storage layer. We showed in earlier publications how our graph-based search over the linked data allows for novel types of queries, including structure queries, statistics, search for simulation setups etc [HWW14]. The data stored in Masymos is cloned from BioModels Database [LD+10] and from the CellML Model Repository [LL+08]. We also indexed the definitions of all bio-ontology concepts used in any of the models for subsequent ranked retrieval. The dominant ontologies are the *Gene Ontology*, the *Chemical Entities of Biological Interest* (ChEBI) and *UniProt*. Additionally, we integrated the Systems Biology Ontology, SBO [CJ+11], and the KInetic Simulation Algorithm Ontology, KiSAO [CJ+11].

CombineArchive Toolkit: Facilitating the transfer of simulation studies. Models need to be shared. The above standards guarantee interoperability of the encoded data, and the concept behind Masymos ensures easy access to simulation models and associated data. However, Masymos focuses on the management of files and on their long-term availability. It does not offer user-friendly data export. An archive-based tool for exporting and then sharing all files belonging to one scientific study is the CombineArchive Toolkit (CAT)[3]. It follows the COMBINE archive specification [BA+14] and generates zip-like file bundles.

The COMBINE archive is a zip container for files related to a simulation study. It uses the Open Modeling EXchange Format (OMEX) [BA+14] which mainly consists of a manifest file, metadata files, and all files necessary to reproduce a virtual experiment. The manifest lists all files belonging to the archive. It stores the format of each file, using a URI, and the file's location within the archive. In addition, files can be flagged "master", meaning those files should be evaluated first, whenever the archive is being loaded into a software tool. The metadata files may contain information encoded in common formats such as RDF, vCard, or Dublin Core Metadata. A list of tools that support the COMBINE archive is available from the specification [BA+14].

The COMBINE Archive Toolkit (CAT) is one implementation of the COMBINE archive format. It was built to support researchers in creating and exploring COMBINE archives. The CAT consists of a core library, a desktop application, and a web-based interface. The CombineArchiveLibrary is implemented in Java and offers all necessary methods to handle COMBINE archives (extract files, browse archive, add or remove files, rename or reorganise files, attach and retrieve metadata). The CombineArchiveWeb interface [SW+14] enables collaborative creation and modification of COMBINE archives. It offers RESTful services and can be used from other client applications. Currently, the CombineArchiveWeb interface has simple connections to BioModels Database and the CellML Model Repository, making it possible to retrieve models. The software is openly available and thus own instances can be installed if particular privacy interests apply.

[3] `https://sems.uni-rostock.de/projects/combinearchive/`

3 Implementation

Our software M2CAT enables researchers to retrieve models or simulation studies from Masymos and directly generates COMBINE archives from all relevant files (Figure 1). The data can then be either downloaded or opened in the CombineArchiveWeb interface.

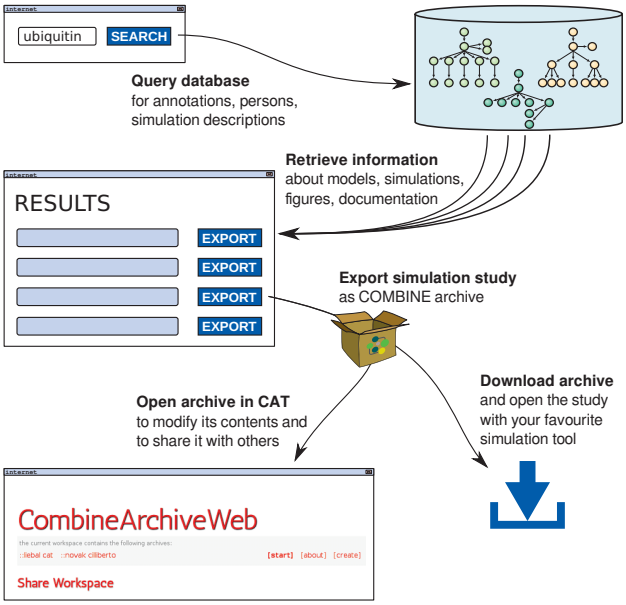


Figure 1: **The M2CAT workflow.** The keywords are used to search model files, simulation setups, and author names in Masymos. The search results are presented in a web interface. They are organised by model name, and displayed together with the additional resources. By clicking the *export* button, users can either download the COMBINE archive of an entry, or open the archive directly in the CombineArchiveWeb tool.

The workflow is implemented in a web interface at m2cat.sems.uni-rostock.de. Masymos can be queried via a simple search field. The keywords are translated into Cypher queries and sent to Masymos via Neo4j's RESTful service. First, M2CAT consults the annotation index, which contains all terms occurring in the semantic annotations of models (Query 1). The result is a list of models and associated documents.

```
START   res=node:annotationIndex('RESOURCETEXT:(ubiquitin)')
MATCH   res<-[rel:is]-(:ANNOTATION)-->(s)-[:BELONGS_TO]->(m:MODEL)-[:BELONGS_TO]->
        (d:DOCUMENT)
RETURN  distinct(m),d
```

Query 1: Return models `m` which contain elements annotated with *ubiquitin* and associated documents `d`.

Afterwards, M2CAT consults Masymos' Person nodes, which contain the names of all authors of reference publications, curators, and other contributors of model code (Query 2). The result is again a list of models and associated documents.

```
MATCH   (d:DOCUMENT)-[HAS_MODEL]->(l:MODEL)-[HAS_ANNOTATION]->
        (k:ANNOTATION)-[HAS_PUBLICATION]->(m:PUBLICATION)-[HAS_AUTHOR]->
        (n:PERSON{FAMILYNAME:"ubiquitin"})
RETURN  distinct(m),d
```

Query 2: Return models `m` which are annotated with publications written by *ubiquitin* and associated documents `d`.

Finally, M2CAT searches for additional resources for each entry in both lists. For example, Query 3 retrieves all simulation descriptions that are linked to a certain model document.

```
START   known=node(273)
MATCH   (known)-->(:MODEL)<--(:SEDML_MODELREFERENCE)<--(s:SEDML)<--(d:DOCUMENT)
RETURN  s,d
```

Query 3: For the document with id *273*: Return simulation descriptions `s` and associated documents `d`.

The sets of documents belonging to the simulation studies and which matched the keywords are then presented on the website. For each set M2CAT provides the model name and a list of available resources.

Using M2CAT it is now possible to export simulation studies as COMBINE archives. Masymos does not store the files themselves, but provides links to all necessary files through the retrieved document nodes. M2CAT resolves these links, retrieves the files, and utilises the CombineArchiveLibrary to create the archive. Specifically, the files are packed, the archive and its files are annotated with metadata about the contents and the creator, and the manifest file is written. The user can then either download the archive or open it in the CombineArchiveWeb tool (see again Figure 1). Downloaded archives simplify journal submissions, if the model code is required together with the manuscript. Opening the COMBINE archive in our web tools enables modellers to add further files to the archive, such as additional figures or supplemental descriptions. Further use cases of the COMBINE archive include easy sharing of files among collaborators, encapsulating datasets used for model development and validation, sharing consistent instances of a model, enabling automatic (machine-only) transfer of research results [BA+14].

4 Summary

The systems biology community has identified reusability of simulation studies as one major challenge in the field. We present here a workflow that combines existing tools for model management and provides a user-friendly interface for downloading reproducible simulation studies. The queried database Masymos already integrates several resources for model-related data. M2CAT gathers this data and generates COMBINE archives from it. Our instance of Masymos currently contains models in SBML and CellML formats and associated simulation experiments in SED-ML format. We think that the usefulness of the COMBINE archives generated by M2CAT increases, if more model-related data are included. Specifically, we wish to include graphical representations of models in SBGN format and simulation results. However, the described workflow is applicable to model repositories in general. It reveals its full power, if the model-related data are already linked, as demonstrated by Masymos.

References

[BA+14] F. Bergmann, R. Adams, et al. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC bioinformatics*, 15(1):369, 2014.

[BDR+10] S. Bechhofer, D. De Roure, et al. Research objects: Towards exchange and reuse of digital knowledge. *The Future of the Web for Collaborative Science*, 2010.

[CJ+11] M. Courtot, N. Juty, et al. Controlled vocabularies and semantics in systems biology. *Molecular systems biology*, 7(1), 2011.

[CVW14] J. Cooper, J.O. Vik, and D. Waltemath. A call for virtual experiments: accelerating the scientific process. *Progress in Biophysics and Molecular Biology*, 10.1016/j.pbiomolbio.2014.10.001, 2014.

[HB+11] M. Hucka, F. Bergmann, et al. A profile of today's SBML-compatible software. In *2011 IEEE 7th International Conference on e-Science*, pages 143–150, 2011.

[HF+03] M. Hucka, A. Finney, et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[HWW14] R. Henkel, O. Wolkenhauer, and D. Walthemath. Combining computational models, semantic annotations, and simulation experiments in a graph database. *DATABASE*, accepted for publication, 2014.

[LD+10] C. Li, M. Donizelli, et al. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC systems biology*, 4(1):92, 2010.

[LL+08] C.M. Lloyd, J.R. Lawson, et al. The CellML model repository. *Bioinformatics*, 24(18):2122–2123, 2008.

[LNH+09] N. Le Novère, M. Hucka, et al. The systems biology graphical notation. *Nature biotechnology*, 27(8):735–741, 2009.

[PSA11] F. Prinz, T. Schlange, and K. Asadullah. Believe it or not: how much can we rely on published data on potential drug targets? *Nature reviews Drug discovery*, 10(9):712–712, 2011.

[SW+14] Martin Scharm, Florian Wendland, et al. The CombineArchiveWeb Application - A Web-based Tool to Handle Files Associated with Modelling Results. In *Proceedings of the 7th International Workshop on Semantic Web Applications and Tools for Life Sciences, Berlin, Germany, December 9-11, 2014.*, 2014.

[WA+11] D. Waltemath, R. Adams, et al. Reproducible computational biology experiments with SED-ML-the simulation experiment description markup language. *BMC systems biology*, 5(1):198, 2011.

[WB+14] D. Waltemath, F. Bergmann, et al. Meeting report from the fourth meeting of the Computational Modeling in Biology Network (COMBINE). *Standards in Genomic Sciences*, 9(3), 2014.

[WO+11] K. Wolstencroft, S. Owen, et al. The SEEK: a platform for sharing data and models in systems biology. *Methods Enzymol*, 500:629–655, 2011.

Genome sequence analysis with MonetDB: a case study on Ebola virus diversity

Robin Cijvat[1] Stefan Manegold[2] Martin Kersten[1,2] Gunnar W. Klau[2]
Alexander Schönhuth[2] Tobias Marschall[3*] Ying Zhang[1,2]
[1]MonetDB Solutions, Amsterdam, The Netherlands
[2]Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
[3]Saarland University & Max Planck Institute for Informatics, Saarbrücken, Germany
[1]*first.last*@monetdbsolutions.com [2]*first.last*@cwi.nl [3]marschal@mpi-inf.mpg.de

Abstract: Next-generation sequencing (NGS) technology has led the life sciences into the big data era. Today, sequencing genomes takes little time and cost, but results in terabytes of data to be stored and analysed. Biologists are often exposed to excessively time consuming and error-prone data management and analysis hurdles. In this paper, we propose a database management system (DBMS) based approach to accelerate and substantially simplify genome sequence analysis. We have extended MonetDB, an open-source column-based DBMS, with a BAM module, which enables *easy*, *flexible*, and *rapid* management and analysis of sequence alignment data stored as Sequence Alignment/Map (SAM/BAM) files. We describe the main features of MonetDB/BAM using a case study on Ebola virus genomes.

1 Introduction

Next-generation sequencing (NGS) technology has confronted the life sciences with a 'DNA data deluge' [SL13]. Thanks to its massively parallel approach, NGS allows for generating vast volumes of sequencing data, which, in comparison to conventional 'first-generation' sequencing methods, happens at drastically reduced costs and processing times. Consequently, biologists now need to invest in the design of data storage, management, and analysis solutions. Ever more often, improvements in this area are no longer an option, but a pressing issue.

As per common NGS-based "re-sequencing" workflows, short DNA fragments are sequenced and subsequently aligned to a reference genome, which aims at determining the differences between the sequenced genome and the reference genome. Thereby, the resulting alignment data files, most often stored in Sequence Alignment/Map (SAM) format or its binary counterpart BAM [L+09], quickly reach the terabyte mark. Complementary software libraries, e.g., SAMtools [L+09], provide basic functionality, such as predicates-based data extraction. For more complex data exploration, scientists usually resort to writing customised software programs.

*This work was done when the author worked at the Life Sciences group of Centrum Wiskunde & Informatica.

However, the traditional file based approach has several drawbacks. First, it does not make use of technology that specifically addresses large data file handling. Even with compression, a BAM file containing the alignment data of a single human genome amounts to hundreds of gigabytes, which, when processing collections of genomes, quickly reaches the terabyte scale [F+14]. Existing file-based tools usually only work properly with data that fits in main memory. Thus, researchers are often left with the non-trivial tasks of partitioning data into optimally fitting pieces and constructing final results from partial results. Moreover, having to repeatedly reload such large files is undesirable. Second, software development and maintenance are extremely time-consuming and error-prone tasks. They require highly specific knowledge of programming languages and applications. Finally, performance is paramount for big data analysis. Therefore, scientists have been increasingly enforced to become "hard-core" programmers, to exploit the full computational power of modern hardware, such as multicore CPUs, GPUs, and FPGAs.

Although Hadoop systems have recently gained much interest in big data processing, they are no ideal candidates for data analysis in life sciences. Hadoop systems are primarily designed for document-based data processing [DG04]. They can be extremely fast in executing simple queries on large number of documents, e.g., distributed grep. But Hadoop systems quickly suffer from serious performance degradation, when they are used to process more complex analytical queries that often involve aggregations and joins [P+09].

The problems mentioned above have been extensively tackled by database management systems (DBMS), which are designed and built to store, manage, and analyse large-scale data. By using a DBMS for genome data analysis, one can significantly reduce the data-to-knowledge time. A major advantage of using a declarative language such as SQL is that the users only need to state *what* they want to analyse, but not *how* exactly to analyse. The DBMS should take care of efficient execution of the queries, e.g. choose the best algorithms, optimise memory usage, automate parallel execution where possible, and make use of the aforementioned modern hardware, while hiding the heterogeneity of underlying hardware and software systems. In this way, scientists can reap the fruits of 50+ years work of the database community on optimising query processing, so as to spend more time on their primary research topics.

However, so far DBMSs have not been widely adopted in the life sciences beyond metadata management. This is mainly due to the mismatch between what existing DBMSs support and what data analysis in life sciences needs. There is a general lack of DBMS functionality and operations to directly query genomic data already stored in files. The current common practice is to first convert the data into CSV files, then load them into a DBMS. This conversion step not only incurs a high data-to-query time, but also substantially increases storage requirements, especially if the original data are compressed. Moreover, it is extremely difficult to keep duplicate data consistent, when there are updates. Finally, although genomic data are encoded in strings, a standard DBMS data type, they have particular semantics. Without dedicated functions, it is not trivial to express even the basic operations on genomic data using SQL, e.g., compute the length of an alignment.

MonetDB/BAM. Our first step towards a solution for big data analysis in life sciences is to tackle the aforementioned functional mismatches. Therefore, we have extended the open-

source column-based DBMS MonetDB[1] with a BAM module[2], which allows *in-database* processing of SAM/BAM files. The software is available as of the Oct2014 release of MonetDB. MonetDB is primarily designed for data warehouse applications, such as data mining and Business Intelligence [M+09]. These applications are identified by their use of large data sets, which are mostly queried to provide business intelligence or decision support. Similar applications also appear frequently in the big data area of e-science, where observations are collected into a warehouse for subsequent scientific analysis. This makes MonetDB a good candidate to provide a data management solution for such applications.

The main features of MonetDB/BAM include: a) SQL loading functions to load a single, a list or a repository of SAM or BAM files; b) SQL export functions allow writing query results to SAM formatted files; c) SQL functions for elementary operations on sequence alignment data, e.g., computing reverse complements of DNA strings and the actual lengths of mapped sequences; and d) automatically construct primary or secondary read pairs, which accelerates analyses on paired alignments. In this paper, we demonstrate how MonetDB/BAM can be used to facilitate genome sequence alignment data analysis, by conducting a case study on a current and highly important topic: studying the diversity of the Ebola virus.

Related work. There are several prototypes that also use DBMSs for genome data analysis. For instance, Röhm et al. [RB09] propose a hybrid data management approach, which relies on file streaming features of SQL Server 2008 to process gene sequence data stored as binary large objects (BLOBs). When faced with large data file, loading data on-the-fly will suffer from the same performance problems as the file-based approaches. Moreover, this work does not consider additional DBMS functionality to facilitate genomic data analysis. Schapranow et al. [SP13] describe an in-memory DBMS platform, HIG, in which existing applications are incorporated for genome analysis. But HIG does not integrate the analysis functionality *into* the DBMS. The work Dorok et al. [D+14] is most closely related to our work, in the sense that it proposes both a DBMS schema to store genome alignment data and an integrated user-defined function `genotype` (in MonetDB) to enable variant calling using simple SQL. However, this work mainly focuses at supporting variant calling. With MonetDB/BAM, we try to target at genome data analysis in general.

2 Ebola virus diversity: a case study

Viruses populate their hosts as swarms of related, but genetically different mutant strains, each defined by its own, characteristic genomic sequence. Analysing such "mutant clouds", often called *viral quasispecies*, is of clinical importance, as it explains virulence, pathogenesis, and resistance to treatment. Exploring the composition of sequences and their relative frequencies, the *genetic diversity* of a quasispecies, based on NGS is a current, central issue in virology [B+12, T+14].

We demonstrate how to make easy use of MonetDB/BAM for some helpful steps towards an easy exploration of the genetic diversity of Ebola infections. Although it has recently

[1] `https://www.monetdb.org/`

[2] `https://www.monetdb.org/bam/`

```
CALL bam.bam_loader_repos('/path/to/ebola-bam-repo', 0)                    (Q1)
```

```
SELECT s.value AS refpos,
       bam.seq_char(s.value, al.seq, al.pos, al.cigar) AS seq_char,
       COUNT(*) AS cnt
FROM generate_series(0, 18960) as s
     JOIN (SELECT pos, seq, cigar FROM bam.alignments_all WHERE pos > 0) AS al
     ON s.value BETWEEN al.pos AND al.pos + bam.seq_length(al.cigar)
GROUP BY refpos, seq_char ORDER BY refpos, seq_char                        (Q2)
```

```
SELECT refpos, SUM(cnt) AS cnt FROM positional WHERE seq_char IS NOT NULL
GROUP BY refpos ORDER BY cnt LIMIT k                                       (Q3)
```

```
SELECT refpos - refpos % 1000 AS grp_start,
       refpos - refpos % 1000 + 1000 AS grp_end, AVG(cnt) AS average
FROM coverage GROUP BY grp_start, grp_end ORDER BY average DESC LIMIT k    (Q4)
```

```
SELECT refpos, coverage.cnt AS coverage, diversity.cnt AS diversity,
       CAST(diversity.cnt AS FLOAT) / coverage.cnt * 100 AS diversity_perc
FROM coverage JOIN (
         SELECT refpos, SUM(cnt) AS cnt FROM base
         WHERE seq_char IS NOT NULL AND seq_char <> SUBSTRING(ref, refpos, 1)
         GROUP BY refpos
     ) diversity USING (refpos)
ORDER BY diversity_perc DESC, coverage DESC, diversity DESC                (Q5)
```

Figure 1: Use case queries

been established that Ebola is a highly diverse and rapidly mutating virus [G+14], conclusive insights are yet to be made. In the project, we use BAM files containing sequence fragments from viral quasispecies of the actual (2014) Ebola outbreak in Sierra Leone [G+14].

Preparing and loading data. First, we retrieved 32 files containing Ebola virus genome sequences (SRP045416) from [G+14]. Together they contain 6,786,308 reads and take 390 MB on hard disk. Then, we used the Burrows-Wheeler Aligner [LD09] to align the reads with the Zaire reference string (NC_002549.1) [V+99], resulting in an average mapping rate of 15.6%. The results are stored in 32 BAM files containing an alignment for every read, with a total size of 500 MB.

All BAM files are loaded into a MonetDB database with the SQL query Q1 in Figure 1. The first argument is the path to the repository of BAM files. The second argument chooses the storage schema: 0 for sequential, 1 for pairwise[2].In this paper we only use the sequential schema (Figure 2), a straightforward mapping of alignment records in BAM files.

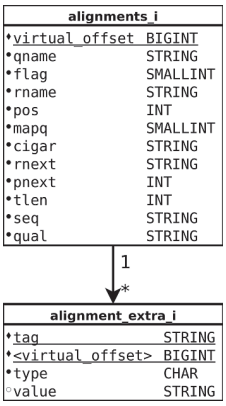


Figure 2: Sequential storage schema

All alignments of one BAM file are stored in two SQL tables `bam.alignments_i` and `bam.alignment_extra_i`, where i is the internal ID of the BAM file. Each tuple in `bam.alignments_i` contains all main fields of one alignment, e.g., `qname`, `flag`, `rname`, `pos`, `cigar`, `seq` and `qual`. The `EXTRA` field of the alignments are parsed and stored in `bam.alignment_extra_i` as `<tag,type,value>` tuples. The `virtual_offset` is used to link the tuples in these tables. For all queries in this work, we have defined a view `bam.alignment_all`, containing `bam.alignment_i` tables of all loaded BAM files.

refpos	seq_char	cnt
...	...	...
46	A	1
46	C	1
47	A	8
...	...	...

Table 1: Result Q2

refpos	cnt
6239	9340
6240	9337
6245	9196
1571	9191
...	...

Table 2: Result Q3

grp_start	grp_end	average
1000	2000	7053.6699999999992
3000	4000	6694.4919999999984
6000	7000	6681.6100000000024
4000	5000	6150.8489999999983
...	...	...

Table 3: Result Q4

Use case 1: computing positional data. Query Q2 in Figure 1 shows how to compute the count of all characters that occur on all positions in MonetDB/BAM. The MonetDB-specific function `generate_series` generates a one-column table with a row for every integer number in the range. We use this in Line 4 to create a table with an entry for every position in the reference genome (NC_002549.1), with a length of 18960 [V$^+$99]. Line 5 selects the alignment position, the sequence, and the CIGAR string for all mapped reads. We join the series table with the mapped reads (lines 4–6). A result tuple is produced if the sequence of the read overlaps with a position in the series table (line 6). The join results are grouped and ordered on the reference positions of the reads and the characters that are found on these positions (line 7). Values of these characters are extracted in the `SELECT` clause (line 2). Finally, from the grouped result, we select the reference positions, the characters on these positions, and their occurrence counters (lines 1–3). Applying Q2 on the Ebola alignment data produces results as shown in Table 1, which reveals that, e.g., on position 46, there is one aligned read with an A and one with a C.

Use case 2: computing coverage and diversity. Assume that the results of Q2 are stored in a table `positional`, query Q3 in Figure 1 shows how to create a top-k of positions that have the highest coverage, i.e., the highest number of aligned reads that overlap with these positions. The results of Q3 in Table 2 show that the reference position 6239 has the highest number of overlapping aligned reads, i.e., 9340.

Assuming the result of Q3 is stored in a view `coverage`, a next step is to calculate a top-k of regions with the highest average coverage, as Q4 in Figure 1. The results of Q4 are in Table 3, which shows that the region 1000–2000 has the highest average coverage.

Diversity is another interesting analysis we can do with the results of Q2 and Q3, i.e., compute the percentage of alignments that differ from the reference genome on each position. The query Q5 in Figure 1 produces a list of positions with their coverage and diversity, with decreasing diversity percentages. In Q5, we have loaded the reference genome string (NC_002549.1) [V$^+$99] in the SQL variable `ref`. The function `SUBSTRING` returns a single character at the given `refpos`. Q5 first computes similar intermediate data as in Q3 (lines 4–6), except filtering out the positions with matching characters with the reference genome (line 5). Then, we join the `coverage` table with the just computed diversity information on the reference position (lines 3–7). This gives us for every position: i) the total number of overlapping read alignments, and ii) the number of overlapping read alignments that differ from the reference genome. Finally, we select the reference position, the count of both the coverage and the diversity subresults, and calculate the diversity percentage for all reference positions as

refpos	coverage	diversity	diversity_perc
721	1471	1471	100
7029	1469	1469	100
5639	1131	1127	99.6463307
...	...	...	...

Table 4: Result Q5

the number of differing read alignments divided by the total number of read alignments for these positions (lines 1,2). The results of Q5 are in Table 4, which e.g. shows that all aligned reads at reference positions 721 and 7029 differ from the reference genome.

Query performance. We run all five queries on a moderate laptop (i7 Quad Core CPU, 4GB RAM, SSD disk). Figure 3 shows the query execution times on different data sizes. The x-axis denotes both the number of files and the file size for each data set. All results are averages of 10 runs.The execution times show a linear behaviour with growing data size. Loading (Q1) and computing positional data (Q2) are the most time consuming tasks. Q1 spends most time on decompressing the BAM files and passing values. The execution times of Q2 include the time to store its results in the `positional` table, which serves as a preprocessing step for the remaining queries. Once data are loaded and preprocessed, the further analysis queries (Q3 – Q5) are done in milliseconds, and the execution times are hardly affected by growing number of files and data sizes.

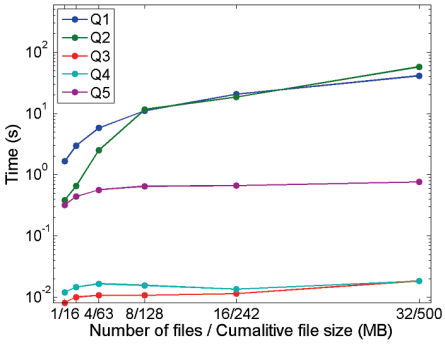


Figure 3: Execution times of all queries

Conclusion and Outlook. In this paper, we showed how to use MonetDB/BAM to facilitate exploration of the genetic diversity of the Ebola virus. Our study indicates that many conceivable analyses on genome sequence alignment data can be easily expressed as SQL queries, provided the DBMS offers the proper functionality. However, before we come up with a comprehensive solution for big data analysis in life sciences, plenty open issues call for consideration. For instance, the performance and scalability of MonetDB-/BAM must be extensively evaluated and improved. We should stress the system with both BAM files of larger genomes, such as human or plant genomes, and terabytes scale file repositories. Also, we should compare the performance of our approach with existing analysis tools, such as BEDTools [AI10]. Moreover, the use cases study should be extended with more important analysis, such as variant calling [N+11], so as to determine more functional requirements MonetDB/BAM should satisfy. Finally, workflow support is a must for scientific exploration. MonetDB already provides various client connections (e.g., JDBC, ODBC), a REST interface, and seamless integration with the R project for statistical computing[1]. Therefore, MonetDB can be easily integrated into existing workflow systems, such as Taverna [W+13].

References

[AI10] Quinlan AR and Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, 2010.

[B+12] N. Beerenwinkel et al. Challenges and opportunities in estimating viral genetic diversity from next-generation sequencing data. *Frontiers in Microbiology*, 2012.

[D+14] S. Dorok et al. Toward Efficient Variant Calling Inside Main-Memory Database Systems. In *DEXA Workshops*, pages 41–45, 2014.

[DG04] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI*, 2004.

[F+14] L. C. Francioli et al. Whole-genome Sequence Variation, Population Structure and Demographic History of the Netherlands. *Nature Genetics*, 46:818–825, 2014.

[G+14] S. K. Gire et al. Genomic surveillance elucidates Ebola virus origin and transmission during the 2014 outbreak. *Science*, 345(6202):1369–1372, 2014.

[L+09] H. Li et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25, 2009.

[LD09] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*, 25:1754–60, 2009.

[M+09] S. Manegold et al. Database Architecture Evolution: Mammals Flourished long before Dinosaurs became Extinct. *PVLDB*, 2(2):1648–1653, 2009.

[N+11] R Nielsen et al. Genotype and SNP calling from next-generation sequencing data. *Nat. Rev. Genet.*, 12(6):443–451, 2011.

[P+09] A Pavlo et al. A Comparison of Approaches to Large-Scale Data Analysis. In *SIGMOD*, 2009.

[RB09] Uwe Röhm and Jose A. Blakeley. Data management for high-throughput genomics. In *CIDR*, 2009.

[SL13] M. C. Schatz and B. Langmead. The DNA data deluge. *IEEE Spectrum*, 50(7):28–33, 2013.

[SP13] Matthieu-P. Schapranow and Hasso Plattner. HIG - An in-memory database platform enabling real-time analyses of genome data. In *BigData*, pages 691–696, 2013.

[T+14] A. Toepfer et al. Viral quasispecies assembly via maximal clique enumeration. *PLoS Computational Biology*, 10(3):e1003515, 2014.

[V+99] V. E. Volchkov et al. Characterization of the L gene and 5' trailer region of Ebola virus. *The Journal of general virology*, 80(Pt2):355–62, 1999.

[W+13] K. Wolstencroft et al. The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Research*, 41(Web Server issue):W557–W561, 2013.

RightInsight: Open Source Architecture for Data Science*

Ahmet Bulut

Department of Computer Science
İstanbul Şehir University
Altunizade Mah. Oymaci Sok. No:15
34660 Uskudar, Istanbul
ahmetbulut@sehir.edu.tr

Abstract: We give the details of our reference architecture called *RightInsight* for enabling rapid data science. RightInsight is based purely on open source technologies. The data is stored in a standard distributed file system such as HDFS. The stored data is processed in Apache Spark, which provides an enhanced Map/Reduce programming environment. Its rich and powerful machine learning base makes it easy to construct descriptive, prescriptive, and predictive models. In addition to providing an agile environment for making sense of the data and the data science problem at hand, its Python-based middleware with a wide array of scientific libraries such as scipy, numpy, matplotlib, and pandas, enables interactive and exploratory data analysis. The ability to ask questions, especially the right questions, and to do what-if analysis is extremely important for any serious data science project. The results of such exploratory analyses are stored in a suitable format that is easily consumable in the Web tier. Using rich JavaScript libraries such as data driven documents and bootstrap, the formatted data can be visualised within a Web browser in creative ways for rapid insight discovery.

1 Introduction

Data scientists are inquisitive. They explore the problem space, ask questions, do what-if analysis, and while doing so they question their existing assumptions and processes. Rather than looking at data from a single source, they examine data from multiple and disparate data sources. All incoming data is sifted through with the goal of discovering hidden insights, which in turn can be used as a competitive business advantage or can be used towards coming up with solutions to pressing business problems. With the necessary analytics and tools support, a capable data scientist will be well equipped to communicate informed conclusions and recommendations across the whole organization [IBM14b].

The necessary analytics and tools support varies. From clustering and regression, to classification and probabilistic inference, and to data enrichment and visualisation, data scientists need to have a solid foundation in computer science and applications, modelling,

*This project is funded by Turkish National Science Foundation (Tubitak) under grant numbers 113E243 and 113E622.

statistics, analytics and mathematics. In order to explore exabytes of data and do what-if analysis, data scientists require powerful back-end systems so-called data science platforms in order to digest raw data. The platforms have to provide an interactive mode of data analysis required due to the iterative and inquisitive nature of data science projects.

Figure 1: The collaborative nature of data science projects, which require business analysts, business users, data scientists, and IT working together for a common objective.

A recent research study conducted by Ventana Research revealed that having the right tools are the most essential for predictive analytics [Res13]. Many organizations have made the necessary human capital investment by hiring experts, who have the core responsibility of ensuring success in predictive analytics. 70% of these companies with business foresight emphasize the need for tools with enhanced usability, being workflow-driven, and ability to handle any information source. The emphasis on usability reiterates the collaborative nature of data science projects, which require business analysts, business users, data scientists, and IT working together for a common objective as shown in Figure 1.[1]

[1] The depicted picture is drawn using the following source images at:
http://www.montel.com/en/markets/business/manufacturing/manuf-factor-it-department
http://blogs.sap.com/innovation/analytics/data-scientist-sexiest-job-century-01242806
http://www.computing.co.uk/IMG/768/231768/datascienceman-370x229.jpg
http://www.phoenixit.im/business_user.html
http://enterprise-dashboard.com/2009/07/
http://thebusyba.com/business-process-modeling-for-new-business-analysts/

1.1 Organization of the paper

The details of RightInsight architecture is discussed in Section 2. In Section 3, we present a representative data science project that exemplifies the primary uses cases for RightInsight. Since data science projects generally require collaboration of multiple stakeholders from multiple departments, i.e., being interdisciplinary in nature and scope, we present two models of interaction that describe how to collaborate in Section 4. Finally, we discuss our current agenda in Section 5 and emphasize key points to take away in Section 6.

Figure 2: The RightInsight architecture for big data science depicted as a workflow.

2 RightInsight Architecture

RightInsight is architected for enabling rapid data science. The architecture is shown in Figure 2. RightInsight is built on open source technologies. Structured, semi-structured, or unstructured data are all stored in a standard distributed file system such as HDFS. The stored data is processed in Apache Spark [ZCF+10], which provides an enhanced Map/Reduce programming environment [DG08]. In order to construct descriptive, prescriptive, and predictive models using the available data, Spark's rich and powerful machine learning base called MLBase is utilised [KTD+13]. On top of it, a Python suite that includes such widely used scientific libraries as scipy, numpy, matplotlib, and pandas enables interactive and exploratory data analysis and the associated model analysis [vR97]. The ability to ask questions and do what-if analysis is extremely important for most data science projects. The agility of the underlying Python-based platform expedites the process of making sense of the data. The results of exploratory analyses made are stored in JavaScript object notation (json). Using a rich set of JavaScript libraries such as data driven documents [BOH11] and bootstrap [Ler12], the information in json can then be visualised inside a Web browser in creative ways for discovering insights and for furthering team communication and collaboration.

There are other computing platforms available for enabling data science. StarCluster is the closest to RightInsight. StarCluster is designed to automate and simplify the process of building, configuring, and managing compute clusters on Amazon's EC2 cloud [MIT13]. A StarCluster suits the computing needs of most distributed and parallel computing applications. The primary difference between RightInsight and StarCluster is the base platform used for scalable data processing. For a StarCluster, OpenMPI, Hadoop, and NFS are used as the base distributed data processing platform [GFB+04]. On the other hand for RightInsight, Spark ecosystem provides a superior performance in processing big data files. More specifically, Spark outperforms Hadoop by up to 80 times in iterative machine learning and graph applications [XRZ+13]. In both RightInsight and StarCluster, a Python suite is used as an interactive shell for agile data analysis.

In Spark, jobs are responsible for analytic or iterative computing on a large dataset. Each job first loads its working data into memory for enabling rapid data access. The main component of Spark is the construct of a resilient distributed dataset (RDD). An RDD provides granular fault tolerance and distribution of work among multiple worker nodes. The original input data is sliced into multiple chunks so that multiple jobs can be created for execution in parallel on each chunk. Fault tolerance is supported through the lineage information stored for a compute flow in the framework per RDD. In case of node failures, each compute step in the compute flow can be re-executed linearly in order to recover. In order to synchronise compute nodes when necessary for certain iterative end-user tasks, RightInsight uses broadcasting feature of Spark for in-flight data and uses Tachyon for data at-rest in contrast to the use of NFS in StarCluster. Tachyon is an in-memory based distributed file system, which enables memory-speed file sharing across cluster frameworks, and which acts as an intermediate layer between HDFS and Spark [LGZ+14].

IBM Watson Analytics is one of the most compelling data science enablers in the industry [IBM14a]. Watson Analytics brings together IBM Cognos Business Analytics capabilities

and IBM SPSS Modelling and Statistics functionalities in order to capture the most relevant facts, patterns and relationships in data, and presents them in an engaging analytics experience to the end users. The primary difference between RightInsight and Watson Analytics is that Watson is built on IBM's proprietary software stack; whereas RightInsight is built on top of an open source technology stack. Similar to the use of Hadoop in StarCluster, Watson uses a proprietary version of Hadoop called IBM BigInsights for improved performance. To the best of our knowledge, there are no benchmarks that compare Spark's performance with IBM BigInsights performance on competitive workloads.

3 Representative Data Science Application

In IBM's Smarter Planet initiative, RightInsight is being proposed as a candidate framework for holistic event analysis in a trillion giga big data ecosystem. Our planet continuously gets smarter and more complex. Heterogeneous information sources on the planet produce streams of events. These information sources could be external sensors (embedded in cars, home appliances, city roads, pipelines, medicine and livestock), internal systems in use, and we the people. Each event is associated with numeric or textual data, or other richer media. From 2005 to 2010, over 33 billion RFID tags were produced, which helped to grow the digital universe from 130 Exabyte (EB) to 800 EB. The digital universe will double every two years till 2020. In 2020, it will be 40 trillion gigabytes, which is more than 5,200 gigabytes for every man, woman, and child in 2020 [GR12].

In a trillion giga big data ecosystem, the collected events may not make sense when evaluated independently but they could indicate much larger developments and/or evolutions when evaluated holistically. Our overarching hypothesis is that there is a higher order generative process that produces the observed events, and our goal is to identify and estimate the parameters of that process through holistic event analysis. Once the inherent generative process is identified, it can be used for testing existing assumptions and doing what-if analysis. For instance, if a harmful development is at work, a security policy can be enforced in order to counteract it. The planned discovery process can be used for measuring the efficacy of the policies enacted and for driving new actionable insights. For holistic event analysis, streams of events have to be monitored for discovering associations, inferring generative themes, and capturing structural relationships. As an example, causal relationships can be captured using Conditional Random Fields (CRFs) [RH13] that are primarily used for structured prediction. Generative themes can be inferred using Latent Dirichlet Allocation (LDA) that is used for learning probability distributions [Bul14]. In this project, we will focus on news events available on the Web. Since news events could arise from disparate data sources, the data processing infrastructure have to be capable of handling the heterogeneity in data. Furthermore, having large amounts of data to work with prompts us to use Apache Spark on top of a Hadoop layer for scalable data analytics.

Holistic data analysis is extremely relevant to IBM's core business [TH11]. The capability to move from reaction to prediction is identified as one of the core business strengths for organizations. Organizations need to interpret and respond to streaming sentiments and dialogues among consumers for taking predictive actions rather than reactive actions.

That is, there is a shift towards increasing the information responsiveness of companies of all sizes. RightInsight can turn data into dollars through holistic event analysis. The generative process learned from the data can be used for predicting the near future or for simulating what can happen in the near future within a short look-ahead window.

At the beginning of 2014, we received funding from National Research Council of Turkey in order to develop a system called BOSS for inferring latent themes in dynamically evolving text collections. There is abundant textual data that evolves over time: Facebook status updates, product comments on e-commerce websites, articles on news sites, tweets, and business transactions to name a few. The goals in BOSS project is in line with the goals in holistic event analysis since inferred latent themes correspond to semantic information that is captured over raw data. And as such, the higher-level semantics helps decision makers drive actionable insights.

4 Collaboration Models for Data Science

Bringing together multiple stakeholders from multiple disciplines and backgrounds is not a straightforward undertaking. Viable collaboration environments and processes need to be crafted for increasing productivity and interaction while not hindering the participation of any stakeholder. We propose two candidate collaboration models for this purpose: (1) OODA and (2) Active Learning.

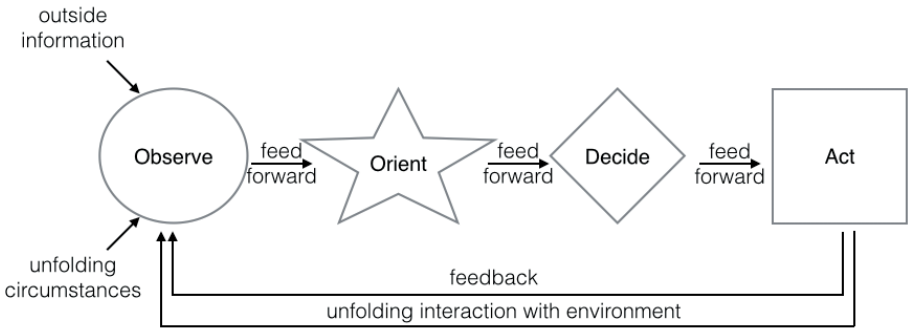


Figure 3: Observe, Orient, Decide, Act (OODA) Loop. (Adapted from http://upload.wikimedia.org/wikipedia/commons/3/3a/OODA.Boyd.svg).

4.1 Observe, Orient, Decide, and Act

In today's highly competitive business environment, all companies that want to stay ahead of the curve have to shorten the time it takes from the point of an observation being made to the point when an action is taken based up on that observation. This is how systems get

smarter, how organizations get more agile, and more importantly how people win.

The time it takes from the point of an observation being made to the point when an action is taken based up on this particular observation is the realization and one cycle in the Observation (O), Orientation (O), Decision (D), and Action (A) loop, i.e., OODA loop as shown in Figure 3 [Boy95]. For most types of decision-making processes in the business world, OODA presents a viable execution model.

RightInsight can facilitate the OODA collaboration model in data collection, analysis of the collected data, and in decision-making through controlled experimentation [KLSH09]. The process was proposed for enabling data-driven policy making in smart cities [Bul12]. It is depicted in Figure 4. The more loops are completed, the more patterns are revealed. Such patterns are catalysts for engineering automated drivers that correspond to these patterns and for crafting expert systems.

Figure 4: Six step process to orchestrate OODA loops.

Using RightInsight, we can detect anomalies using custom monitors, diagnose fault by comparing operational data with baselines, and finally apply control based on diagnosis [HY06]. This whole process describes a single OODA loop. The actions that are taken based on the decisions given at the last step affect the underlying social and physical processes. These processes give rise to a totally new set of observations. This in turn marks the beginning of the next OODA loop.

4.2 Active Learning

Imagine the following scenario that describes how a historian and a computer scientist collaborates. By using RightInsight, the computer scientist cultivates from a heterogeneous set of digital information sources prepares a portfolio of “interesting” events. The event

sources are spatio-temporal. At the beginning, the collected events are seemingly interesting events. Furthermore, the number of events presented to the historian is limited. The historian evaluates the suggested events and identifies which events are indeed interesting and may be worth delving deeper into. Armed with this insight, the computer scientist goes back to cultivation and drills in further for coming up with more interesting events to present. This process continues indefinitely with one caveat. Since the historian is a precious resource with limited time and energy, the computer scientist and the system in turn has to actively learn from the historian.

"Active Learning" defines the interdisciplinary collaboration between a computer scientist and a social scientist. Technically, active learning is used for training classifiers with less training data than needed during a regular supervised training [Set09]. The key idea behind active learning is that when the learning algorithm is allowed to choose the data from which it learns, then it can perform up to par with less training data. This is valuable in situations where unlabelled data is abundant but labelling them is expensive.

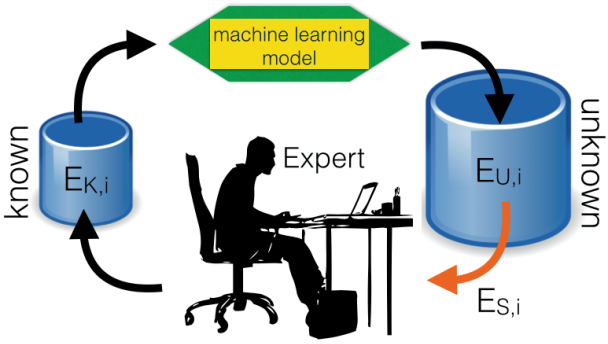


Figure 5: Illustration of an active learning cycle.

In our sample scenario, a custom data selector that is built based up on the historian's domain expertise can speed up the learning process with less training data. RightInsight provides the necessary tools suitable to interact with data and enables iterative machine learning for easing exploratory analysis outlined in the scenario. Let E be the total set of all events including those events that are known to have a certain interesting characteristic. During each active learning step i as depicted in Figure 5, the set E is broken up into three subsets as known events $\mathbf{E}_{K,i}$, unknown events $\mathbf{E}_{U,i}$, and $\mathbf{E}_{S,i} \subset \mathbf{E}_{U,i}$ that is selected for expert opinion. There is an active body of research on the best method to use for event selection. All events that carry a certain characteristic that are known to exist in previously known interesting events can be chosen for expert opinion. However this naive approach does not explore parts of the event space that is yet to be explored. The events to select is an interplay between the exploration and the exploitation over the event space representation. Thomson sampling, which is an implementation of exploration and exploitation tradeoff, has been shown to work well for display advertisement selection and news article recommendation [CL11]. Thomson sampling is used for event selection in RightInsight.

5 Discussion

RightInsight architecture is instrumental for us to set up a center of excellence in data science. The center stands on the shoulders of human resources and compute resources. Academicians and data scientists are the key human resources. A masters program in Data Science, which is a joint venture with IBM Turkey, sustains the human resources pipeline and supports the ongoing skills development. The compute resources are provisioned among various in-house data science projects. The initial infrastructure is an elastic private cloud; a hybrid or a purely public cloud will be considered when necessary.

6 Conclusions

RightInsight is an enabler of data science. For agility, it is built on top of open source technologies. Heterogenous data is stored in a standard distributed file system, and then processed iteratively in Apache Spark. Complex descriptive, prescriptive, and predictive models can be built on the collected data using a rich and powerful machine learning base called MLBase. On top of the base data processing platform, a Python suite enables interactive & exploratory data analysis, and the associated model analysis. The agility of the Python-based suite expedites the process of making sense of the data. The results of exploratory analyses made are visualised in a Web browser in creative ways for discovering insights and for furthering team collaboration.

References

[BOH11] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, December 2011.

[Boy95] J. Boyd. The Essence of Winning and Losing. In *United States Air Force*, 1995.

[Bul12] A. Bulut. *City Competitiveness and Improving Urban Subsystems: Technologies and Applications*, chapter City Competitiveness and Infrastructure, pages 1–20. IGI Global, November 2012.

[Bul14] A. Bulut. TopicMachine: Conversion Prediction in Search Advertising Using Latent Topic Models. *IEEE Transactions on Knowledge and Data Engineering*, 26:2846 – 2858, November 2014.

[CL11] O. Chapelle and L. Li. An Empirical Evaluation of Thompson Sampling. In *25th Neural Information Processing Systems (NIPS)*, pages 2249–2257, December 2011.

[DG08] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *ACM Communications*, 51(1):107–113, January 2008.

[GFB+04] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B.Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall. Open MPI: Goals, Concept, and Design of a Next Generation MPI

Implementation. In *11th European PVM/MPI Users' Group Meeting*, pages 97–104, September 2004.

[GR12] J. Gantz and D. Reinsel. Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. In *IDC iView sponsored by EMC*, December 2012.

[HY06] T. Han and B. S. Yang. Development of e-maintenance system integrating advanced techniques. *Computers in Industry*, 57:569–580, 2006.

[IBM14a] IBM. Watson Analytics. In *http://www.ibm.com/analytics/watson-analytics/*, 2014.

[IBM14b] IBM. What's a data scientist. In *http://www-01.ibm.com/software/data/infosphere/data-scientist/*, 2014.

[KLSH09] R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne. Controlled Experiments on the Web: Survey and Practical Guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, 2009.

[KTD+13] T. Kraska, A. Talwalkar, J. Duchi, R. Griffith, M. J. Franklin, and M. Jordan. MLbase: A Distributed Machine-learning system. In *6th Biennial Conference on Innovative Data Systems*, 2013.

[Ler12] R. M. Lerner. At the Forge: Twitter Bootstrap. *Linux Journal*, 2012(218):6, 2012.

[LGZ+14] H. Li, A. Ghodsi, M. Zaharia, S. Shenker, and I. Stoica. Tachyon: Reliable, Memory Speed Storage for Cluster Computing Frameworks. In *ACM Symposium on Cloud Computing*, November 2014.

[MIT13] MIT. StarCluster. In *http://star.mit.edu/cluster/*, 2013.

[Res13] Ventana Research. Deploying Predictive Analytics for Competitive Advantage. In *http://www.slideshare.net/VentanaResearch/ventana-research-infographicpredictiveanalyticsibm2013final*, 2013.

[RH13] K. Radinsky and E. Horvitz. Mining the Web to Predict Future Events. In *6th ACM International Conference on Web Search and Data Mining*, pages 255–264, 2013.

[Set09] B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[TH11] M. Teerlink and M. Haydock. Customer analytics pay off. In *IBM Institute for Business Value*, 2011.

[vR97] G. van Rossum. Scripting the Web with Python. *World Wide Web Journal*, 2, 1997.

[XRZ+13] R. S. Xin, J. Rosen, M. Zaharia, M. J. Franklin, S. Shenker, and I. Stoica. Shark: SQL and Rich Analytics at Scale. In *ACM International Conference on Management of Data (SIGMOD)*, pages 13–24, 2013.

[ZCF+10] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. In *2nd USENIX conference on Hot topics in Cloud Computing*, pages 10–10, 2010.

Rethinking Spatial Processing in Data-Intensive Science

Christian Authmann, Christian Beilschmidt,
Johannes Drönner, Michael Mattig, Bernhard Seeger

Department of Mathematics and Computer Science
University of Marburg, Germany
{authmanc, beilschmidt, droenner, mattig, seeger}@mathematik.uni-marburg.de

Abstract: In this paper we address the problem of supporting data-driven science in geo-scientific application domains where scientists need to filter, analyze, and correlate large data sets in an interactive manner. We identify ten fundamental requirements for such a new type of processing system. They span from supporting spatial data types over using workflows for data lineage, to fast and flexible computations for exploratory analysis. Our study of related work looks at a range of established systems of different domains and shows significant drawbacks with respect to our requirements. We propose the **V**isualization, **A**ggregation & **T**ransformation system (VAT) as an architecture to overcome current limitations. It bases on standard software and implements performance-relevant parts using state-of-the-art technology like GPU computing. Our first comparison with other systems shows the validity of our approach.

1 Introduction

The management of data has become a critical issue in challenging problems of mankind related to climate change and biodiversity. An example is the current dramatic loss of species on earth. The Task Group on Data and Scenario Support for Impact and Climate Analysis of the Intergovernmental Panel on Climate Change (IPCC), which is the leading international body for the assessment of climate change, is dedicated to identifying the information needs in global research projects on climate impacts, adaptation, and mitigation. In the context of biodiversity, governments have given substantial financial support to build up large data centers for archiving and processing to enable scientists identifying the key indicators of species loss and deriving accurate models about biodiversity.

Our work is motivated by two ongoing biodiversity projects with different focus. GFBio[1] [DGG+14] is an integrative German platform to provide a sustainable, service oriented, national data infrastructure facilitating data sharing and stimulating data-intensive science in the fields of biological and environmental research. Our task in the project is to provide added-value services and data management tools for analysis and visualization. IDESSA[2] is a research project that addresses the real problem of sustainable rangeland management in South African savannas. In both projects, we have to deal with very large heterogeneous data and scientists interested in extracting hidden information in an exploratory way.

[1] www.gfbio.org
[2] www.idessa.org

As exemplified in our projects, processing of spatial data has already met the three key properties – Variety, Volume and Velocity – of Big Data, even before the term was coined recently [MCB+11]. **The data is heterogeneous (Variety).** Researchers have to deal with varying data formats, coordinate reference systems, resolutions and uncertainties [BSE+12]. There are raster images which arrange the earth's surface in a regular grid where each cell contains a measurement value. They represent continuous phenomena like elevation, temperature or precipitation. There are point data sets that associate attributes to locations of interest, e.g. occurrences of a species or measurements of a sensor. In addition, there are line and polygonal data sets for modelling areas of interest like roads and the habitat of species. **The data is big (Volume).** A global raster with a cell size of 100×100 meters exceeds 100 gigabytes of data. Current satellite instruments like MODIS send 70 gigabytes of data per day, with the next generation of satellites increasing this by an order of magnitude. **Fast and responsive data processing (Velocity)** is difficult to achieve with traditional methods such that response times meet the expectations of interactive users. This is especially important for exploratory analyses and refinement.

The use of traditional tools on spatial data confines scientists to a limited set of questions that can be answered. Several approaches have been proposed to modernize the tools by applying Big Data principles to spatial data. We show the architecture of an integrated system, a holistic approach to vector and raster data, tiled processing, low-resolution previews and the efficient usage of modern hardware architectures.

The paper is structured as follows. It first examines requirements for an interactive exploratory research system in Section 2. Section 3 addresses related technologies and describes their shortcomings with respect to the requirements. The main part in Section 4 introduces our own approach for overcoming current limitations. The evaluation in Section 5 shows the validity of our approach by experimentally comparing it to other approaches.

2 Requirements

Scientists that are approaching emergent scientific challenges in biodiversity research face a diverse set of tasks to accomplish. We identified the following criteria (R1-R10) as mandatory for a system that effectively supports biodiversity science.

A support for raster images, point and polygon data is obligatory (R1). In addition to the geographical information, temporal information is inherent for nearly all data items. They either have a point in time of creation or a validity interval which must be considered for semantically sound computations. This leads to the necessity of a model of time and temporal semantics (R2). Data from different sources often has different formats and reference systems. A system should provide automatic data transformation techniques that allow matching data nonetheless (R3). It should recognize the need for such operations and apply them transparently to take the responsibility from the user.

The system has to support a rich set of low-level *operators* which perform well-defined transformations on data (R4). Point and polygon intersection, and filtering points by their attributes are two examples of such operators. To achieve higher level functionality, users

can combine these operators into a *workflow* (R5). A workflow consists of a set of operators and their parameters, as well as a specification of the data flow between these operators. They are often modeled and visualized as directed graphs.

A standardized workflow format allows re-using, sharing and publishing of workflows (R6). Since a workflow contains all processing steps from raw measurements to the scientific results, they can provide *data lineage*. This is the traceability of the involved data. Publishing a workflow allows other scientists to verify the calculations and reproduce the results. Moreover, users are interested in downloading results of the computation in a desired format and use them within their preferred working environment.

Scientific research is often an iterative approach of forming an idea, testing a hypothesis, reviewing the results and forming a new idea which starts the iteration again. To support fast iterative research, the system must support flexible composition and modification of workflows. Results must be available in a near real-time fashion (R7).

With growing data sizes, it is increasingly expensive for researchers to keep local copies of relevant data and perform computations on their workstations. All data should thus be stored and processed on a central system which also provides commonly used data sets (R8). Researchers must also be able to upload custom data (R9). An intuitive and efficient Web interface shall allow researchers to explore data and generate workflows (R10).

3 Related Work

In this section we review existing systems and software components in whether they fulfill the aforementioned requirements. The traditional way to implement a workflow is to program its functionality manually. Scripting languages like R[3] offer functionality to load and process spatial data. We now look at systems from different categories like GIS, scientific databases, workflow systems and Big Data solutions.

Geographic Information Systems (GIS) are usually desktop applications with a rich user interface which allow loading, visualization and processing of spatial data. Systems like QGIS[4] offer a vast set of low and high level operators. Users however often need to unify the format of their input data manually beforehand (R4). A few GIS implementations offer basic lineage information and rudimentary workflows, but not to the extent required for exploratory data analysis (R7). As they are installed on a user's local machine they are limited to the available resources. Most current GIS implementations are not designed to handle data that exceeds a computer's main memory.

Recently more and more cloud GIS become available. They overcome the limitations of local hardware, but are mostly highly specialized applications that focus e.g. on data visualization rather than processing. One example is Map Of Life[5] that utilizes the Google Earth Engine[6]. However, it is difficult to extend these systems for other use cases (R9, R10).

[3] www.r-project.org
[4] www.qgis.org
[5] www.mol.org
[6] earthengine.google.org

Spatial data frameworks are on the other end of the spectrum. They offer low level functionality but are not feasible for end users as standalone systems (R10). Representatives are GDAL[7] for reading, writing and re-projecting spatial data and GEOS[8] for vector operations. They are very useful in a backend that hides the complicated usage from the user. However they do not exploit recent hardware developments sufficiently well (R7).

Scientific databases for spatial data focus on storing and processing rather than on visualization. There are specialized systems for handling large rasters like RasDaMan [BDF+98] and SciDB [Bro10]. While they provide scalability, they lack support for vector data and other spatial data types (R1). PostGIS[9], as an extender for PostgreSQL, is much more of a general purpose system. Being based on a classical RDBMS, it lacks the expressiveness and performance required for spatial data analysis (R5, R7). It is suitable for storing and retrieving spatial data, not for analytical processing.

Scientific workflow systems [CVD+12, VER+15] are very prominent in the field of biodiversity informatics. They allow the creation of a process by specifying a data flow using a visual drag and drop editor. Taverna[10] is one of the most popular systems. It allows to orchestrate Web services and manages the data transmissions. Users can share workflows on platforms like BiodiversityCatalogue[11]. Others can then incorporate them e.g. as sub workflows in their own processes. Thus there is a lot of functionality already available. However, users are bound to the availability of external Web services and have limited control over changes of the services. This hinders reproducibility of computations (R6). Additionally Web services have protocol overhead and the orchestration of multiple distributed services leads to high network traffic. This causes slow execution of larger workloads and hinders the exploratory usage of such systems (R7).

In the context of Big Data, Hadoop [Lam10] is omnipresent. Hadoop-GIS extends Hadoop with GIS features [AWV+13] which could make it a good fit for our scenario. It is however a batch processing framework. Long start-up times [DG08] and constant disk usage for intermediate results hinder our desired near real-time computation of workflows (R7).

Concluding our survey we did not find a single system that matches all of our requirements. A combination of the key features of the above mentioned systems is necessary to create an interactive processing system.

4 The VAT System

Our goal is to develop a system that meets the ten requirements of Section 2. This section gives a brief overview of our concepts and the state of development we started in 2014. Firstly, it introduces the architecture of the system and sets the scope for the rest of the section. Secondly, it describes our core design decision regarding the functionality and performance of the system.

[7] www.gdal.org
[8] geos.osgeo.org
[9] www.postgis.net
[10] www.taverna.org.uk
[11] www.biodiversitycatalogue.org

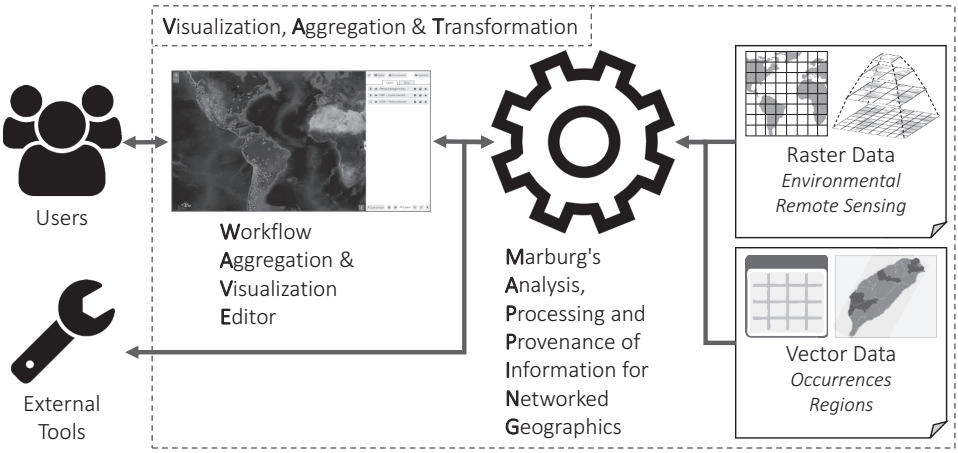


Figure 1: Overview of the architecture of the VAT system.

Figure 1 depicts the architecture of our **V**isualization, **A**ggregation & **T**ransformation system (VAT). It consists of two main parts. The back end, called MAPPING, is responsible for managing and processing of data. The front end, called WAVE, is a web-based application for visualization and manipulation of data. It allows the definition of workflows. WAVE facilitates the interactive usage of MAPPING, but is not in the scope of the rest of this section. The producer side (right) shows the integration of different data types like raster and vector data from presumably different sources. The consumer (left) is either a user that communicates via WAVE or an external application that can interact with MAPPING directly via Web services and programming interfaces.

4.1 Functionality

VAT is an integrated system that allows us to have full control of each individual processing step. It follows a holistic approach and supports both raster and vector data. We store rasters in a read-efficient way using a custom engine with tiling and pyramids of different resolutions. Different levels of compression allow us to specify trade-offs between CPU time and storage space. We currently use PostGIS for the storage of vector data.

We make use of GDAL and GEOS for import, export and low-level operations whenever applicable. The system implements several standard operators as known from GIS programs, e.g. for the correlation of data from different types. Future work will include more operators, both for general and specialized applications.

MAPPING represents workflows internally as a directed tree which WAVE is able to visualize for comprehensibility. The result of the workflow is the root node, while the leafs represent the inputs. When one operator consumes the result of another operator that has a different coordinate reference systems than expected, VAT introduces a new re-projection

operator that is plugged in automatically. This converts the output of the source operator such that the sink operator is able to use it. In the future it will be possible to perform parameter sweeps by automatically repeating computations for parameter values in a certain range. An example is to perform a computation for every month in a given year.

The result of a workflow must be reproducible in order to support the scientific method. This means if a user runs a workflow again at some point, it must yield the same results. The first step is to store workflows such that they can be re-used. Additionally, both data and operators have to be versioned. This way, users can execute workflows on old data with old versions of the operators, retrieving the same results as in the original execution. The system can recompute all results at any point in time and does not necessarily have to store them. To encourage sharing and collaboration, it will be possible to deliver workflows to others and also make use of existing workflows.

Standard-compliant interfaces are very important. We support import and export over standard OGC[12] protocols, like Web Map Service (WMS), Web Coverage Service (WCS) and Web Feature Service (WFS). This allows using our system as part of a larger workflow if desired. A researcher might use our system for computationally heavy tasks, then export the results to a local machine for analysis with specialized tools.

4.2 Performance

Section 2 disclosed that an exploratory approach highly benefits the research process. This makes fast response times a key aspect in getting users to accept the VAT system for their purposes. Performance is orthogonal to the functional requirements, but influenced our design decisions. We focus on optimizing performance on a single machine first. The distribution of workload across multiple machines will be addressed in our future work.

In general, only a subset of the data is needed for answering a query, e.g. if the region of interest is restricted to Germany. This is especially crucial in the context of rasters because of their size. We employ optimization techniques of current database technologies to optimize the processing and minimize the size of temporary results. To avoid reading unneeded data we push the region of interest in form of a query rectangle down to the source operators. This is comparable to the selection push-down in the relational database world. We can exploit this to constraint the data transmission as early as possible. In cases where different coordinate reference systems are involved, we transform the area of interest into other coordinate systems as required.

As mentioned in Subsection 4.1 we store pre-aggregated versions of rasters in pyramids. We exploit them to compute on lower resolutions. This accelerates the exploratory processing significantly, giving a researcher often enough information to decide whether to follow up on an idea. Users can pan and zoom on their map, allowing to calculate results only for the currently viewed area in the screen's resolution. When an interesting correlation is found, results can be computed in full detail.

[12] www.opengeospatial.org

From a performance perspective it is not desirable to move large data over the network to perform computations on it. Under the assumption that the gain in computational power is negligible in contrast to I/O costs, it is more reasonable to process the data where it resides. Our approach is therefore to choose function shipping over data shipping to minimize network traffic. We allow users to upload scripts (currently we support R) to the server, where they are executed close to the data. This allows users to implement custom processing steps without downloading the data and processing locally. The system executes a script by an operator as part of a workflow, allowing lineage and reproducibility even on workflows with custom scripts.

In order to enable fast and efficient data processing, it is essential to exploit a single machine's resources effectively. On modern hardware this means to parallelize work as much as possible to incorporate all CPU and GPU cores. GPUs are predestined for processing rasters, even though current GIS do not use them extensively. To combine CPU based operators with GPU accelerated ones, we are looking at Heterogeneous System Architecture[13] (HSA), where CPU and GPU share a common main memory address space. This avoids expensive copies of data from one memory space to the other.

5 Evaluation

To demonstrate the validity of our approach, we designed a use case for the processing of spatial data and compared ease of use as well as processing time in different systems. The use case is a plausible real-world scenario suitable for benchmarks and comparisons. We do not claim its scientific relevance – that is a question for biodiversity researchers.

5.1 Description of the Use Case

The goal of our use case is to find possible habitats for a species, in this case the European wildcat (*Felis silvestris silvestris*). According to the IUCN/SSC Cat Specialist Group[14], they have three requirements on their environment: 1) They avoid areas with human activity and prefer forests. 2) Areas with heavy snow cover are avoided. 3) They have been observed up to 2250m.

To produce a map of the possible habitats we used environmental data from *WorldClim* [HCP+05] and the *Global Landcover Map for the year 2000* (GLC2000[15]). Figure 2 shows the example workflow. We used GLC2000 to look up human activity and forests, and classify the area according to the first criterion. The snow cover is more difficult to calculate. Without access to global snow cover measurements, we approximate the snow cover using WorldClim's temperature and precipitation data. Since WorldClim has monthly aggregated data, the snow cover is calculated separately for December and January. The sum of

[13] www.hsafoundation.com
[14] catsg.org
[15] bioval.jrc.ec.europa.eu/products/glc2000/glc2000.php

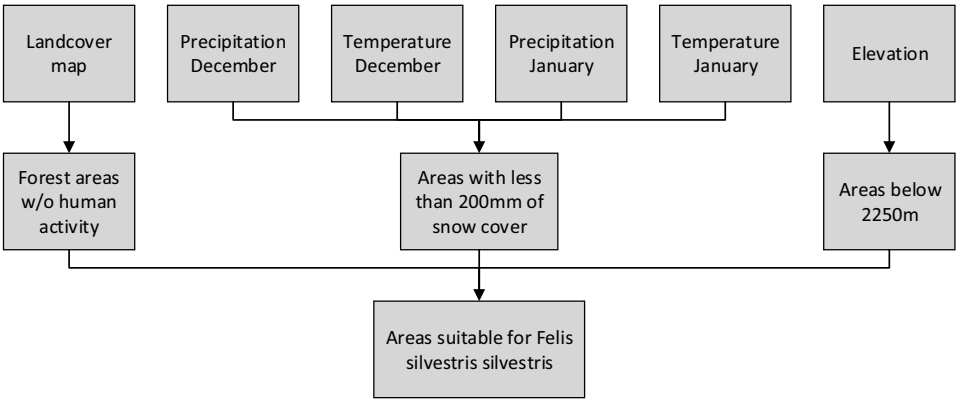


Figure 2: Workflow to find suitable habitats for wildcats

both values is compared to a threshold (below 200mm) to classify the area according to the second criterion. The third criterion uses the WorldClim elevation model to discard all areas with an elevation above 2250m. In a last processing step, we merge the classifications into a single result, showing the areas where all three classifications match.

5.2 Implementation

To facilitate comparison, we restricted the use case to raster data and transformed the data into the same size and coordinate system. While all tools should be able to handle different data types and coordinate systems, implementing the required conversions in multiple systems proved to be too laborious.

R: The *raster* package from R allows loading and processing of rasters. We implemented the workflow as a single-step computation. The multi-step computation implemented on all other systems proved to be too slow. The implementation of the computation required experience in R, but was otherwise straightforward.

QGIS: While QGIS itself provides a set of processing tools, it relies heavily on other GIS to use their algorithms. We used the *r.mapcalc* module from GRASS[16] to apply formulas to each pixel. The integrated graphical modeling tool was used to implement a workflow containing all steps of the computation. The visual tools were easy to use. The workflow was quick to implement and worked out of the box.

Taverna: Taverna is an orchestrator and requires external web services to handle the actual computations. We installed pyWPS[17] with GRASS as the processing back end. To avoid being limited by the network, the Web service was running on the same machine as Taverna.

[16] grass.osgeo.org
[17] pywps.wald.intevation.org

	MAPPING	QGIS	Taverna	R
Germany	0.7s	3.5s	53.1s	48s
Europe	3.8s	35s	-	3600s
World	29s	368s	-	18600s

Table 1: Measurement results of the use case experiment

The implementation using Taverna proved to be challenging. Setting up the Web service required manual programming and configuration. Implementing the workflow in Taverna was met with several difficult to debug errors, from XML encoding problems to out-of-memory errors. We managed to work around some of these problems, but were unable to run the workflow on the full data sets.

MAPPING: In our own system, all computation steps were implemented using built-in operators. The *expression* operator works similar to GRASS's *r.mapcalc* and was used for the majority of the computation. Our visual editor allows an easy implementation of the workflow, similar to QGIS. GPUs have not been exploited for this comparison.

5.3 Results

We ran all tests in a virtual machine on a consumer PC with an SSD and 16 GB of main memory. Table 1 shows the run time of the workflow execution on differently sized areas. The rasters for Germany are 1200×1200 pixels in size. The ones for Europe are roughly 40× bigger than for Germany and the whole world again is 12× bigger than Europe. The results show substantial improvements of MAPPING in comparison to its competitors. For Germany MAPPING is between 5× and 75× faster than the other systems. The difference is even more apparent for larger regions.

6 Conclusion and Future Work

We presented a general architecture and selected key design decisions for a system to solve emerging computational requirements in the spatial research domain. Our experiments showed the potential speed up our system is able to achieve even in the early state of development. This makes our approach interesting for interactive research on large data. However the system is currently not distributed and not scalable to any amount of data.

In our future work we will achieve scalability in our system by distributing data and computation across multiple machines. A geographical partitioning will allow splitting up a query and computing it on multiple regions in parallel. A disjoint partitioning may result in heavy network traffic if neighbor data is needed. Storing overlapping regions (cf. [Bro10]) helps diminishing this problem. Different regions could vary heavily in data volume and request load. An example is an unexplored sea-region in contrast to a national park. In this

sense it is important to choose regions according to these factors. Additionally, it is useful to log the activity to have a better prediction for workloads.

Since workflows have a well-defined and never-changing result, it is possible to cache both intermediate and final results of workflows. Caching these results will speed up execution of different workflows with identical parts. We will examine existing caching solutions for their suitability for geo-aware caching.

Our evaluation is only able to give a first impression of the performance of systems in one representative but also specific use case. To reduce the bias in such comparisons, there is a demand for a unified benchmark. It has to capture the variety of data and workloads scientists encounter in their everyday research in the context of spatio-temporal data. This facilitates the development and evaluation of systems that are used for supporting scientific research questions.

References

[AWV+13] A. Aji, F. Wang, H. Vo, et al. Hadoop-GIS: A High Performance Spatial Data Warehousing System over MapReduce. Proc. of the VLDB Endowment, Vol. 6(11), 2013.

[BDF+98] P. Baumann, A. Dehmel, P. Furtado, et al. The Multidimensional Database System RasDaMan. In *ACM SIGMOD Conference*, SIGMOD '98, pages 575–577. ACM, 1998.

[Bro10] Paul G. Brown. Overview of sciDB: Large Scale Array Storage, Processing and Analysis. In *Proc. of the ACM SIGMOD Conference*, pages 963–968. ACM, 2010.

[BSE+12] K. Bach, D. Schäfer, N. Enke, et al. A comparative evaluation of technical solutions for long-term data repositories in integrative biodiversity research. *Ecological Informatics*, 11:16–24, 2012.

[CVD+12] V. Cuevas-Vicenttín, S. Dey, et al. Scientific Workflows and Provenance: Introduction and Research Opportunities. *Datenbank-Spektrum*, 12(3):193–203, 2012.

[DG08] J. Dean and S. Ghemawat. MapReduce. *Comm. of the ACM*, 51(1):107, 2008.

[DGG+14] M. Diepenbroek, F. Göckner, P Grobe, et al. Towards an Integrated Biodiversity and Ecological Research Data Management and Archiving Platform: The German Federation for the Curation of Biological Data (GFBio): Informatik 2014 – Big Data Komplexität meistern. GI-Edition: LNI(232):1711–1724, 2014.

[HCP+05] R. J. Hijmans, S. E. Cameron, J. L. Parra, et al. Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology*, 25(15):1965–1978, 2005.

[Lam10] C. Lam. *Hadoop in Action*. Manning Pub., Greenwich, CT, USA, 1st edition, 2010.

[MCB+11] J. Manyika, M. Chui, B. Brown, et al. *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. McKinsey Global Institute, 2011.

[VER+15] C. Vitolo, Y. Elkhatib, D. Reusser, et al. Web technologies for environmental Big Data. *Environmental Modelling & Software*, 63:185–198, 2015.

Studierendenprogramm

CBIR gestütztes Gemälde-Browsing

Marc Büngener

Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
Universitätsstraße 1
40225 Düsseldorf
marc.buengener@uni-duesseldorf.de

Abstract: Zur Beantwortung der Frage, warum Bildersuchen auf Schlüsselwörtern basieren, statt auf den bildimmanenten Informationen, werden Repräsentationsmöglichkeiten für Bildinformationen vorgestellt, die Personen in die Lage versetzen, Anfragen in Form von Bildinformationen statt Schlüsselwörtern zu formulieren. Die Bildinformationen sollen in einer Weise interpretiert werden, die dem subjektiven Empfinden menschlicher Betrachter entsprechen. Als Bildinformationen werden Farben, Texturen und die Darstellung einfacher Formen ausgewertet.

1 Motivation

Gegenwärtig populäre Bildersuchen bestimmen Ergebnisse oft textbasiert, aufgrund personell zugeordneter Schlüsselwörter.[1] Typische Schlüsselwörter für Gemälde sind der Titel, die Stilrichtung oder der Künstler eines Gemäldes. Doch wie kann beispielsweise in einem Online-Postershop nach Gemälden oder allgemein nach Bildern gesucht werden, wenn diese Attribute unbekannt sind oder die anfragestellende Person nur eine ungenaue Vorstellung von den Eigenschaften eines gesuchten Bildes hat? Individuelle Eindrücke, wie Stimmungen und Gefühle, die ein Gemälde vermittelt, sind schwierig durch Schlagwörter formulierbar. Deshalb wird eine Bildersuche durch textbasierte Annotationen nicht untersucht. Aus exakt messbaren Bildinformationen sollen weniger exakte oder nur subjektiv empfundene Informationen abgeleitet werden. Durch die Beschreibung der Farblichkeit von Bildern soll die von ihnen vermittelte Stimmung indirekt anfragbar werden. Die Beschreibung von Texturen soll die Suche nach Gemälden eines bestimmten Malstils ermöglichen. Durch die Detektion einfacher Formen, die auf einem Gemälde abgebildet sind, soll nach Gemälden mit ähnlichen Motiven gesucht werden können. Es wird davon ausgegangen, dass diese Verknüpfung von bildimmanenten Eigenschaften und Wirkungen auf einen menschlichen Betrachter naheliegen.

Warum basieren Bildersuchen auf Schlüsselwörtern, statt auf bildimmanenten Eigenschaf-

[1] http://www.guggenheim.org, http://sammlung-online.museum-folkwang.de, http://www.nrw-museum.de, http://www.kunstsammlung.de, http://www.wga.hu, http://www.kunstkopie.de, http://www.artcyclopedia.com, http://www.google.de, http://images.google.de, http://de.fotolia.com/, https://www.flickr.com/explore, weitere Beispiele: http://www.at-web.de/bildsuche

ten? Ausgehend von Dattas Artikel *Image Retrieval: Ideas, Influences, and Trends of the New Age* [DJLW08] wurden eigene Lösungsansätze für diese zentrale Frage entwickelt und untersucht.[2] Während Schmitt [Sch04] in seiner Habilitationsschrift CBIR umfassend beschreibt, beschreibt Kästers [Käs05] Dissertation inhaltsbasierte Anfragemöglichkeiten und maschinelles Lernen durch Benutzerfeedback besonders genau. Mit der Kenntnis bestehender Methoden wurden eine Suche nach Bildern von Kunstgemälden von Neuem unternommen, von denen die anfragestellende Person nur eine vage Vorstellung hat.

2 Gemälde-Browsing

Analyse Ein Mensch, dem ein Bild beschrieben wird, bezieht die Beschreibung auf seine eigene Erlebniswelt oder diejenige, welche er mit der beschreibenden Person verbindet. Eine Beschreibung wird in einem Kontext verstanden. Ein automatisches System ist grundsätzlich nicht in der Lage, eine kontextabhängige Interpretation der Beschreibung zu leisten. Wie kann die Vorstellung einer Person von einem Bild einem automatischen System als Anfrage gestellt werden? Wie kann eine Person ihre subjektive und zudem lückenhafte oder ungenaue Vorstellung in exakt qualifizierten und quantifizierten Bildeigenschaften ausdrücken? Welches sind die bedeutenden exakt formulierbaren Eigenschaften, welche die Vorstellung der anfragestellenden Person auszeichnen?

Im Alltag beschreiben Menschen untereinander Bilder, indem sie die auf einem Bild dargestellten Objekte und charakteristischen Farben nennen. Für eine detaillierte Bildbeschreibung werden die Anordnung der Objekte auf der Bildfläche beschrieben.

Ablauf der Anwendung Um die Eignung und Qualität der verwendeten bildimmanenten Merkmale direkt in der Anwendung sichtbar zu machen, wird ein einfaches Design gewählt, welches die formulierten Anfragekriterien möglichst wenig interpretiert. Die Gemäldesuche wird dazu in Schritte gegliedert. Abbildung 1 stellt den groben Ablauf der Gemäldesuche schematisch dar. Der erste Schritt ist die Auswahl von Farben, welche für die Beschreibung der gesuchten Gemälde bedeutungsvoll sind. Dem Benutzer wird daraufhin eine initiale Bildermenge angezeigt, die seiner Farbauswahl entspricht. Auf dieser Bildermenge werden durch den Benutzer in einem zweiten Schritt beispielhaft Gemälde ausgewählt, welche den Vorstellungen der anfragestellenden Person nahe kommen. Dieser zweite Schritt ist eine vereinfachte Form von Relevance Feedback und kann beliebig häufig wiederholt werden. Die Bewertung durch den Benutzer beschränkt sich auf die Auswahl als passend empfundener Gemälde. Eine Gewichtung oder Interpretation nicht ausgewählter Gemälde findet nicht statt. Zur Berechnung von Gemälden als Ergebnis einer Anfrage werden feste Parameterwerte verwendet; deren Optimierung ist ein gesondert zu untersuchendes Problem.

[2] vgl. das vom Hermitage Museum St. Petersburg eingesetzte Verfahren *Query By Image Content (QBIC)* der Firma IBM. [FSN+95]

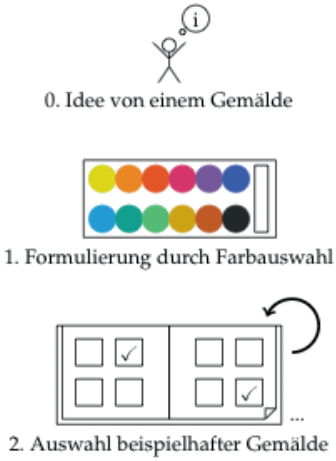


Abbildung 1: Ablauf der Gemäldesuche

2.1 Initiale Bildermenge

Farben können in der implementierten Anwendung durch die Auswahl von Wertebereichen für einzelne Dimensionen des HSV- und des RGB-Farbraumes formuliert werden. Zusätzlich wird eine selbst definierte Farbtafel angeboten, deren Farben feste Wertebereichskombinationen der Dimensionen des HSV-Farbraumes repräsentieren. Die Auswahlmöglichkeiten werden durch Abbildungen der Mittelwerte ihrer Wertebereiche dargestellt. Durch diese bereits inhaltsbasierte Formulierung der Anfrage wird die Notwendigkeit exakter Angaben für vage Vorstellungen vermieden.

Die initiale Bildermenge ist der gesamte Datenbestand, absteigend sortiert anhand der normalisierten Häufigkeitswerte ausgewählter Farbattribute. Eine Farbe wird in einem Gemälde als um so bedeutungsvoller interpretiert, je mehr Bildfläche sie einnimmt. Der Zusammenhang gleichfarbiger Bildflächen oder eine Mindestgröße gleichfarbiger Flächen wird nicht erfasst. Wählt der Benutzer mehr als einen der vorgegebenen Wertebereiche der Farbmodelle oder mehr als eine Farbe der Farbtafel aus, so werden deren normalisierten Häufigkeitswerte addiert. Dadurch werden Gemälde, die mehrere ausgewählte Farbattribute enthalten, potentiell vorrangig sortiert. Zur Übersichtlichkeit werden nicht alle Gemälde gleichzeitig angezeigt, sondern seitenweise in Gruppen zu je 20 Gemälden. Durch eine Änderung der Farbauswahl wird eine neue Suche begonnen. Zur Beschleunigung der Suche werden die normalisierten Häufigkeitswerte der verwendeten Farbmodelle beim Import der Gemälde in die Datenbank vorausberechnet und gespeichert, sodaß sie zur Laufzeit der Anwendung schnell angefragt werden können.

2.2 Ähnliche Gemälde

In den Ergebnismengen der Anfrageschritte kann der Benutzer blättern und Gemälde markieren, die er als seinen Vorstellungen entsprechend *empfindet*. Als nächste Ergebnisse werden daraufhin Gemälde präsentiert, die den ausgewählten Gemälden im Datenbestand am ähnlichsten sind. Wurden mehrere Gemälde ausgewählt, werden zu jedem ausgewählten Gemälde ähnliche Gemälde bestimmt. Die Menge aller ähnlichen Gemälde aller ausgewählten Gemälde wird anhand ihres Ähnlichkeitsmaßes sortiert. Die Ähnlichkeit kann aufgrund der Farbe, Textur oder absoluten Häufigkeiten dargestellter einfacher Formen bestimmt werden.

Für die Ähnlichkeitsbestimmung aufgrund der Farbe wird der Vergleich von Gemälden anhand ihrer Histogrammschnitte bezüglich ihrer Farbhistogramme im HSV- und RGB-Farbmodell und bezüglich der vorgeschlagenen Farbtafel zur Auswahl angeboten.

Für die Ähnlichkeitsbestimmung aufgrund der Gemälde-Textur kann die Dichte interessanter Bildpunkte verwendet werden. Sie wird durch die Anzahl der Keypoints im Verhältnis zur Bildgröße beschrieben. Andererseits kann die Ähnlichkeit der Gemälde-Textur aufgrund des Histogrammschnitts der in Histogrammen erfassten visuellen Wörter berechnet werden. Als interessante Bildpunkte können die für die Keypoint-Dichte mit dem Harris-Laplace-Operator [Lin98] detektierten Keypoints für den Bag of Visual Words-Ansatz wiederverwendet werden.

Für die Ähnlichkeitsbestimmung aufgrund einfacher Formen werden die absoluten Anzahlen in den Gemälden identifizierbarer Vielecke verglichen. Eine Normalisierung ist nicht sinnvoll, weil die Formen als Annäherung an die dargestellten Motive verwendet werden. Ein Bezug der Motive zur Bildgröße ist nicht beabsichtigt.

Zur Beschleunigung der Suche ähnlicher Gemälde werden zu allen Gemälden des Datenbestandes eine feste Anzahl nächstähnlicher Gemälde vorausberechnet und in der Datenbank gespeichert.

2.3 Historienfunktion

Durch eine Historienfunktion wird dem Benutzer ermöglicht, seine Vorstellungen nicht nur zu konkretisieren, sondern sie auch während des Vor- und Zurückblätterns durch die bisher gezeigten Gemälde zu ändern. Durch eine geänderte Auswahl zuvor präsentierter Gemälde, wird die Suche absichtlich *nicht verzweigt* dargestellt. Die Seiten ähnlicher Gemälde einer veränderten Gemäldeauswahl werden an die bisher präsentierten Gemälde angefügt. Die Abbildungen 2 und 3 zeigen schematische Darstellungen beispielhafter Seitenabfolgen. Sie zeigen, welche Navigationsmöglichkeiten bestehen und wie eine Verzweigung durch das Einfügen ähnlicher Gemälde vermieden wird.

Durch die Auswahl von Farben und ähnlichen Gemälden, kann die Anfrage zwar subtil, aber nur grob formuliert werden. Mit dem Ziel, die semantische Lücke zu verkleinern, wird die Formulierbarkeit durch das Browsing in einer Historie ergänzt. Die Historie ist -

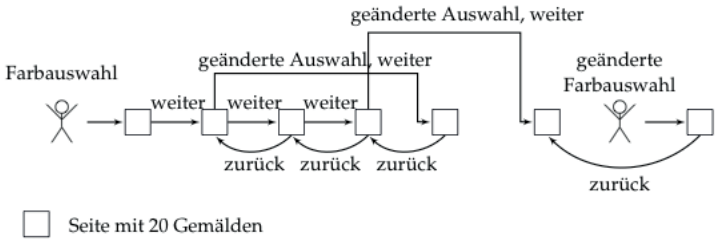


Abbildung 2: Navigationsmöglichkeiten durch Historienfunktion

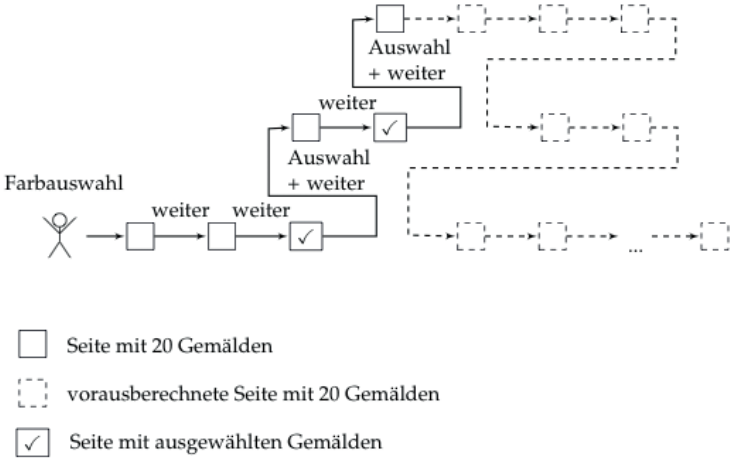


Abbildung 3: Einfügen ähnlicher Gemälde

bewusst vereinfachend - linear aufgebaut, um den Anfragekomfort nicht einzuschränken. Ihre chronologische Anordnung ist ein Aspekt von Relevance Feedback; Relevance Feedback birgt die Gefahr, durch häufige Rückfragen und die Erfordernis von Bewertungen, den Anfragekomfort einzuschränken.

Alternativ zur Historienfunktion wäre ein Clustering ähnlicher Gemälde und ihre Darstellung in einer Weise, die einer vieldimensionalen Mindmap vergleichbar ist. Problematisch sind hierbei die vielfältigen Entscheidungsmöglichkeiten des Benutzers bei der Navigation. Durch die Entscheidung für die Sichtung eines Clusters werden eventuell individuell gesuchte Ergebnisse nicht angezeigt. In einer Anwendung könnten Gemälde in unterschiedlichen Clustern mehrfach dargestellt werden (vgl. Fuzzy-Clustering). Es wird unterstellt, dass eine derart aufwendige Darstellung eine anfragestellende Person überfordert.

3 Bewertung

3.1 Datenmenge

Die Gemäldesuche wird auf der Datenmenge der in der *Web Gallery of Art*[3] enthaltenen Bilddaten durchgeführt. Zum verwendeten Stand Mai 2014 sind dies 24.915 Dateien. Es sind die Dateien, die im Katalog der Galerie als Bilder von Gemälden erfasst sind. Die Gemälde stammen von 2.896 Künstlern aus den Jahren 1001 bis 1900, vorwiegend aus der Epoche der Renaissance. Je Künstler enthält der Datenbestand zwischen 1 und 560 Bilder von Gemälden. Die meisten Gemälde stammen von Giotto di Bondone (560), Vincent van Gogh (330) und Peter Paul Rubens (289).

3.2 Versuchsaufbau, Versuchsstrategie

Die Anforderung einer Bildersuche anhand ungenauer Vorstellungen bedeutet, dass kein eindeutiges Ziel der Suche existiert. Die Erfüllung subjektiver vager Vorstellungen ist nicht objektiv messbar. Es ist lediglich möglich, viele Stichproben subjektiv auszuwerten, ob die berechneten Gemälde den ungenauen Vorstellungen entsprechen, die einer Anfrage zugrunde liegen. Die Eignung eines Algorithmus ist unter zwei Gesichtspunkten feststellbar: *Enthält die berechnete Ergebnismenge ein für Versuchszwecke vorgegebenes Ziel-Gemälde? Enthält die berechnete Ergebnismenge Gemälde, die als unpassend empfunden werden?* Messbare Werte für diese Fragestellungen sind der Rang, auf dem ein vorgegebenes Gemäle in der Ergebnisliste positioniert ist und die Anzahl der im Hinblick auf die Anfragekriterien als unpassend empfundenen Gemälde. Unter der Voraussetzung, dass für alle Messreihen die gleiche Datenmenge verwendet wird, können diese Werte verwendet werden, um die implementieren Methoden zur Bestimmung von Ergebnis-Gemälden zueinander in Beziehung zu setzen. Dadurch können die unterschiedlichen Methoden zumindest vergleichend bewertet werden. Ein absolutes Qualitätsmaß kann nicht abgeleitet werden, weil sowohl der Rang, als auch die Anzahl der als unpassend empfundenen Gemälde, von der Größe der Datenmenge abhängig sind.

Weil die Gemäldesuche in Teilprobleme zerlegt wurde, werden auch die Lösungen der Teilprobleme experimentell getrennt voneinander untersucht.

In den folgenden zwei Unterkapiteln werden die Suche vorgegebener Gemälde und die als unpassend empfundenen Ergebnisgemälde getrennt voneinander beschrieben. Sie gliedern sich jeweils in die Durchführung und Bewertung der Farbauswahl und die Durchführung und Bewertung der Suche ähnlicher Gemälde.

[3] http://www.wga.hu

3.3 Suche vorgegebene Gemälde

3.3.1 Farbauswahl

Zum Test der Farbauswahlmöglichkeiten wurden alle Gemälde des Datenbestandes personell gesichtet. Es wurden Gemälde ausgewählt, nach denen durch die Auswahl einer Farbe gesucht werden können soll. Es wird erwartet, dass die vorgegebenen Gemälde unter den ersten 100 Gemälden einer entsprechenden Farbauswahl präsentiert werden.

Es wurden Anfragen zu allen Farbwinkel-Wertebereichen des HSV-Farbmodells und der selbst definierten Farbtafel durchgeführt. Die Sättigungs- und Helligkeits-Wertebereiche wurden zunächst nicht berücksichtigt. Die Anfrage von Farben über die Auswahlmöglichkeit von RGB-Wertebereichen wurde nicht untersucht, weil diese Art der Formulierung von Farben für den menschlichen Benutzer nicht intuitiv ist.

Bereits bei den ersten Versuchen fiel auf, dass die alleinige Auswahl von Farbwinkel-Wertebereichen zur Anzeige vieler sehr dunkler Gemälde auf den vorderen Ergebnispositionen führt. Deshalb wurden zusätzlich zum untersuchten Farbwinkelbereich die Helligkeitsbereiche eingeschränkt. Es wird unterstellt, dass der Benutzer keine Gemälde minimaler Helligkeit sucht. Die Farbsättigung wurde nicht spezifiziert.

Die Formulierung von Farbe durch Farbmodelle, wie HSV und ganz besonders RGB, sind für einen menschlichen Anwender problematisch. Einerseits kann er nicht alle interessanten Farben - wie beispielsweise Schwarz, Weiß, Grau, Braun oder Fleischfarbe - formulieren, andererseits muss er Farben exakt formulieren, die schwierig unterscheidbar sind - wie Cobalt, Blau und Cyan. Darüberhinaus muss der Anwender die Dimensionen der Farbmodelle in seiner Auswahl geeignet kombinierten. Die Verwendung einer fest vorgegebenen Kombination von Wertebereichen aller Dimensionen eines Farbraum zum Vorschlag einer Farbtafel interessanter Farben löst dieses Problem.

3.3.2 Ähnliche Gemälde

Um die Suche nach ähnlichen Gemälden überprüfen zu können, wurden Stichproben von Gemäldepaaren bestimmt, die subjektiv als ähnlich eingeschätzt werden. Die Einschätzung kann durch eine ähnliche Stimmung, ähnliche Malstile oder ähnliche Motive der Gemälde begründet sein.

Mangels eines objektiven Ähnlichkeitsmaßes für diese subjektiven Eigenschaften wurden die verwendeten Algorithmen zur Bestimmung der Farblichkeit, der Textur und der einfachen Formen eines Bildes relativ zueinander betrachtet. Als Hinweis auf die Eignung eines Vergleichskriteriums wird die Trefferposition in der jeweiligen vorausberechneten Datenbanktabelle nächstähnlicher Gemälde ausgewertet.

Die Gemälde wurden in den Experimenten jeweils gegenseitig gesucht. Im ersten Schritt wurde anhand der Auswahl eines Gemäldes 1, das Gemälde 2 erwartet. Im zweiten Schritt wurde anhand der Auswahl des Gemäldes 2 das Gemälde 1 erwartet. Die Suchrichtung für die paarweise als ähnlich eingeschätzten Gemälde stellte sich in vielen Beispielen als sehr bedeutungsvoll heraus. Dies zeigt die Verletzung der Symmetrieeigenschaft von Distanz-

funktionen durch die verwendeten Ähnlichkeitsmaße.

Die Suche ähnlicher Gemälden anhand der Vergleiche von Keypoint-Dichten und den Anzahlen detektierter einfacher Formen war unbefriedigend. Nur in wenigen Fällen waren diese Methoden überhaupt Trefferpositionen unter 1.000 ermittelbar.

Die Gemäldepaare der Stichproben, bei denen nur die Farbvergleiche gute Ergebnisse erzielten, zeigen zwar ähnliche Motive, aber die Ähnlichkeit der vermittelten Stimmungen überwiegt subjektiv. Die Gemäldepaare der Stichproben, bei denen nur die Texturvergleiche gute Ergebnisse erzielten, sind einander so ähnlich, dass sie tatsächlich die gleichen Motive darstellen. Schlechte Ergebnisse bei Farbvergleichen von Gemälden werden häufig durch Farbverschiebungen infolge unterschiedlicher Belichtungen bei der fotographischen Reproduktion der Gemälde bewirkt. Einige Bilder derselben Gemälde zeigen im Vergleich zueinander beispielsweise eher gelbe oder graue Farbtöne.

Die Ähnlichkeit von Gemälden, bei denen ein Gemälde nur einen Ausschnitt eines anderen Gemäldes zeigt, wird durch einen Vergleich der Farbverteilungen nur unzureichend erfasst. Es zeigt sich jedoch, dass die Ähnlichkeitsbestimmung über die Häufigkeiten von SIFT-Deskriptoren (Scale Invariant Feature Transform [Low04]) im verwendeten Bag of Visual Words-Ansatz in diesen Fällen relativ gut funktioniert. Dies könnte durch die Skalierungsunabhängigkeit der SIFT-Deskriptoren hervorgerufen werden.

Eine Möglichkeit zur Verallgemeinerung, dass ähnliche Gemälde entweder anhand ihrer Farbgestaltung oder ihrer Textur oder erst durch Kombination beider Vergleichsmethoden bestimmt werden können, ist nicht ersichtlich. Beispiele subjektiv ähnlich empfundener Gemälde, die nur als wenig ähnlich berechnet wurden, zeigen bestehende Schwächen des Vergleichs bildimmanenter Eigenschaften. Dies ist ein Hinweis darauf, dass menschliche Betrachter einerseits ein größeres Maß Unähnlichkeitstoleranz erwarten, andererseits objektiv ähnlichere Gemälde subjektiv als weniger ähnlich betrachten. Die Einschätzung von Ähnlichkeit durch einen menschlichen Betrachter wird stark durch eine selektive Wahrnehmung einzelner Gemäldeeigenschaften beeinflusst.

3.4 Als unpassend empfundene Gemälde

3.4.1 Farbauswahl

Um zu prüfen, wie gut die objektive Bilderauswahl anhand der Farbverteilung mit dem subjektiven Empfinden übereinstimmen, wurden für verschiedene Farbauswahlmöglichkeiten jeweils die ersten 100 Gemälde betrachtet.

Für die Farbauswahl durch *Farbwinkel* des HSV-Farbmodells stellte sich deren große Abhängigkeit der Farbwirkung von der Helligkeit heraus. Durch die Auswahl eines Farbwinkelwertebereichs alleine wird ein Farbeindruck nur unzureichend beschrieben. Die Einteilung der Farbwinkel-Wertebereiche ist für einen menschlichen Betrachter zu detailliert. Dies zeigen Ergebnismengen, denen nur im direkten Vergleich ein Farbwinkel zugeordnet werden kann. Diese feine Differenzierung ist für einen Menschen zur Beschreibung von Gemälden aus der Erinnerung oder einer ungenauen Vorstellung heraus nicht möglich.

Die Ergebnismengen aufgrund der Auswahl von Farben der selbst definierten *Farbtafel* sind leichter nachvollziehbar. So zeigt die Ergebnismenge für die Auswahl des roten Wertebereichs keine Schwarz-Weiß-Gemälde, wie bei der Auswahl des roten Wertebereichs für die Farbwinkel des HSV-Farbmodells. Die den Ergebnismengen zugrunde liegenden Farbauswahlen sind leichter unterscheidbar, die Farben sind klarer differenziert. Es fallen nur wenige Gemälde mit unerwarteten Farbfeldern auf.

Das Konzept, Farben aus festen Kombinationen von Wertebereichen der Dimensionen des HSV-Farbmodells zu beschreiben, überzeugt durch die Ergebnisqualität und die leichte Formulierbarkeit von Farben, die nicht alleine durch einen Farbwinkel qualifiziert werden können. Die Farbtafel ist darüberhinaus beliebig spezifizier- und erweiterbar.

3.4.2 Ähnliche Gemälde

Für die Bewertung der Ähnlichkeitsbestimmung von Gemälden zu einem ausgewählten Referenzgemälde wurden für jede implementierte Vergleichsmethode ebenfalls die 100 nächstähnlichen Gemälde betrachtet. Eine Identifikation, warum die berechneten 100 Gemälde die ähnlichsten Gemälde sind, ist wegen der unüberschaubaren Größe des Datenbestandes personell nicht möglich.

Die Ähnlichkeitsbestimmungen anhand der *Farblichkeit* erscheint für alle verwendeten Farbmodelle passend. Aufgrund der unterschiedlichen Genauigkeiten, in denen die Farbwerte der Farbmodelle in Histogrammen erfasst wurden, erscheinen die Ergebnismengen abhängig vom vergleichenden Farbmodell ebenfalls unterschiedlich genau.

Für die Ähnlichkeitsbestimmung anhand der *Textur* zeigt sich der alleinige Vergleich der Keypoint-Dichten als ungeeignet. Die Ähnlichkeit der berechneten ähnlichen Gemälde ist nicht nachvollziehbar. Der Vergleich der Gemäldetexturen über den Bag of Visual Words-Ansatz zeigt jedoch gute Ergebnisse. Zwar ist es auch hier schwierig, einen personellen Vergleich der Textur mit dem Referenzgemälde durchzuführen, die Texturen aller 101 Gemälde erscheinen jedoch sehr ähnlich.

Die Auswertung dargestellter einfacher *Formen* eines Gemäldes, um Gemälde mit ähnlichen Motiven zu berechnen, ist in der implementierten Form nicht aussagekräftig. Die Ergebnisse sind nicht nachvollziehbar. Die Formendetektion ist sehr stark von der Anwendung von Filtern und damit von vielen Parametern und Parameterkombinationen abhängig. In der Implementierung wurde zwar in dem Bewußtsein darauf verzichtet, dass die Optimierung der Parameterwahl eine eigene Herausforderung darstellt. Das Ergebnis ist dennoch unerwartet schlecht.

4 Ausblick

Die erarbeiteten Ansätze zur Formulierung und Berücksichtigung bildimmanenter Merkmale für eine Gemäldesuche bieten viele Möglichkeiten zur Verbesserung, Erweiterung und alternativen Implementierung. Die Wertebereiche der selbst definierten Farbtafel können

genauer untersucht und angepasst werden.

Eine genauere Anfragemöglichkeit für die Farbensuche bietet eine Segmentierung der Gemälde. Dazu müssten lediglich die Farinformationen bestimmter Regionen der Gemälde gesondert erfasst und anfragbar gemacht werden. Dadurch könnten beispielsweise Gemälde angefragt werden, deren obere Hälfte der Bildfläche blau und deren untere Bildhälfte grün ist.

Für die Repräsentation der Textureigenschaften der Gemälde können alternative Keypoint-Detektoren und -Deskriptoren verwendet werden. Statt der SIFT-Deskriptoren werden für die Objekterkennung auf Bildern auch HOG-Deskriptoren (Histogram of Oriented Gradients [DT05]) vielversprechend verwendet.

Um neue ähnliche Gemälde speichern zu können, ohne die Erfordernis einer neuen Vorausberechnung aller jeweils ähnlichen Gemälde, könnte die Verwendung einer Graphdatenbank geeignet sein. Die nächstähnlichen Gemälde müssen dabei nicht in einer relationalen Datenbanktabelle gesondert gespeichert werden. Die Ähnlichkeitsattribute der Gemälde legen die Einordnung in einen Graphen fest, wenn sie in den Datenbestand importiert werden.

Die semantische Lücke kann zwar kontextabhängig verkleinert werden und der Umgang mit ihr kann in Anwendungen komfortabler gestaltet werden, als Problem bleibt sie jedoch bestehen.

Literatur

[DJLW08] Ritendra Datta, Dhiraj Joshi, Jia Li und James Z. Wang. Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Comput. Surv.*, 40(2):5:1–5:60, Mai 2008.

[DT05] Navneet Dalal und Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, Jgg. 1, Seiten 886–893, June 2005.

[FSN+95] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele und Peter Yanker. Query by Image and Video Content: The QBIC System. *Computer*, 28(9):23–32, September 1995.

[Käs05] Thomas Käster. *Intelligente Bildersuche durch den Einsatz inhaltsbasierter Techniken*. Dissertation, Universität Bielefeld, 2005.

[Lin98] Tony Lindeberg. Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[Low04] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[Sch04] Ingo Dr.-Ing. Schmitt. *Multimedia-Datenbanken: Retrieval, Suchalgorithmen und Anfragebearbeitung*. Dissertation, Otto-von-Guericke Universität Magdeburg, 2004.

Musical Similarity Analysis based on Chroma Features and Text Retrieval Methods

David Englmeier[1], *Nina Hubig*[2], *Sebastian Goebl*[1], *Christian Böhm*[1]

1 - University of Munich, 2 - Helmholtz Center Munich
LFE Datenbanksysteme, iKDD-group
{englmeier, hubig, goebl, boehm}@dbs.ifi.lmu.de

Abstract: At the present day the world wide web is full of music. Highly effective algorithms for music compression and high data storage has made it easy to access all kind of music easily. However, it is not possible to look for a similar piece of music or a sound as easily as to google for a similar kind of text. Music is filtered by its title or artist. Although musicians can publish their compositions in a second, they will only be found by high youtube ratings or by market basket analysis. Less known artists need much luck to get heard, although their music might just be what people want to hear. To approach this issue, we propose a new framework called MIRA (Music Information Retrieval Application) for analyzing audio files with existing Information Retrieval (IR) methods. Text retrieval has already yielded many highly efficient and generally accepted methods to assess the semantic distance of different text. We use these methods by translating music into equivalent audio words based on chroma features. We show that our framework can easily match music interpreted even by different artists.

1 Introduction

A modern, effective and personalized music recommendation system needs to focus on multimodal features: musical content and context as well as user correlation [GMS12]. Musical content comprises low-level features acquired by signal processing methods as there are Mel Frequency Cepstral Coefficients or chromatograms, both based on fourier analysis. Others are rate of zeros crossings, upper envelope for loudness, bandwidth analysis or spectral centroids. These low-level features might appear unrelated to music recommendation or melody, but are simple to calculate and, indeed, build a solid base for more sophisticated similarity measures. Mid-level features for musical content apply statistic and machine learning methods on low-level features to gain rythmic information by beat histograms or tonal information as basic melodies or chords. Chroma features are used very frequently. High-level features intend to build user related concepts out of low- and mid-level features. Examples are euphony, infered atmosphere, musical genre or even automatically triggered dance moves. To gain these features, as for text retrieval, semantic features need to be extracted. Additional from physical sound data we need to focus on processes of music perception and reactions to it by various consumers. Insights from physics and psychology, acoustics and cognitive musicology need to be combined to step ahead in music analysis: relations beween sounds and sympathy have to be understood to connect musical content

and recommendation.

Features of musical context comprise information which can not be extracted by music itself. Examples are: meaning of vocals, musician's political background, geographic origin [Dow09]. These features have an influence on perception of a piece of music and, thus, on music recommendation.

User context is also used for music recommendation. It takes search and music listening habits into account. An example for visual aspects are album covers. Music marketing tries to focus covers on potential buyers, although there is no direct connection betweeen music and album art. Choice of music is also a social habit as listeners listen to friends' recommendation. Even larger social groups used musical genres for defining their identity.

For an optimal music recommendation system all three types of features have to be taken into account.

For solving such questions, this paper introduces the framework MIRA. It applies different existing IR-methods, specifically the new Explicit Semantic Analysis (ESA) method to music files. As we focus on chroma features MIRA finds and ranks similar cover versions of a given song title.

The remainder of this paper is the following: First we give some background knowledge on the most interesting applied IR methods in MIRA, Section 3 then shows how these methods are used in the context of music files. The experiment Section 4 shows some results for these methods. Section 5 concludes this paper.

2 Background

This section gives the necessary background information of the most important (1) methods how to extract features from audio files, specifically how to get a chroma feature and (2) to the two most important applied text-retrieval methods for these chroma features.

2.1 Basic methods of feature extraction

The technique in extraction features from audio files strongly determines how the frequency looks like. We introduce the Constant Q Transform and the chroma features that both evolved from the Fast Fourier Transform (FFT).

2.1.1 Constant Q Transform

Constant Q Transform introduced 1990 by Judith C. Brown [Bro90] solves a problem that couldn't be solved by the original FFT algorithm where it bases on: adapting to musical frequencies. The reason is that musical frequencies have no equidistant intervals that are necessary for FFT but act more like links in geometric sequences, in which the quotient of

two closely related links is constant. The interval of two links f_i and f_{i+1} with $f_i = f_0 \cdot 2^{\frac{i}{12}}$ is calculated as follows:

$$\delta_i = f_{i+1} - f_i = f_i(2^{\frac{1}{12}} - 1)$$

As can be seen, the interval depends on the frequence f_i and is not the same for all pairs of frequencies as is the case for DFT.

2.1.2 Chroma features

But even the wide tone spectrum of the Constant Q Transform is due to the large number values not entirely able to be a similarity measure for music pieces. Its values have no meaning for a single music piece. Thus, it is further edited to gain another significant recognition feature: the chroma feature. As found by [GMS12] the chromagram created from the chroma features represents an audio feature that reflects the characteristics of the music piece well and is on the other hand immune to specific attributes of recordings of the same song. To create such chroma vectors all tones of different octave of the corresponding 12 half tones are mapped into one octave. This means that for example tone "A" is added to a value, whose sum represents a component of the chroma vector, regardless of its respective octave. This is regarded as some sort of weight and can be calculated as follows [Lug09]:

$$CV(i) = \sum_{m=0}^{M-1} |X_{CQ}(i + 12m)|$$

where i is the entry index of a respective chroma vector CV, M is the number of octaves and X_{CQ} is the vector calculated by the Constant Q Transform. The standard way to enhance the robustness of a chroma vector is normalization. It makes the CV specifically robust to changes in volume and dynamic of the audio file [GMS12]. These positive effects through postprocessing as well as the robustness make chroma vectors a sensible choice for finding live - or cover-versions of the same song [RHG08]. In Figure 1 you see a chromagram that shows in a time line the different exististing chroma vectors of the classical music piece "For Elise" by Ludwig van Beethoven. The intensity of the tones are colored in either red (high amplitude), green (medium amplitude) to blue (low amplitude).

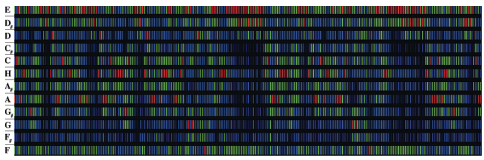


Figure 1: Chromagram: Ludwig van Beethoven, title: „For Elise“

2.2 Applied IR techniques

Here we explain the two most important text- or information retrieval techniques used in this framework.

2.2.1 TF-IDF weights

The TF-IDF weights is a measure which expresses the meaning of a term or a document within a collection of documents. This measure was established in IR by Salton and Yang [SY73] and will be used in this framework on audio words, terms and songs as documents in the respective song collection. The TF-IDF measure is composed of two weights: a local weight depending on the document and a global weight that takes the complete document collection into account.

The local weight $tf_{i,j}$ is based on the frequency of a given term i in a document j and calculated by the following formula [BYRN11]:

$$tf_{i,j} = \frac{h_{i,j}}{\sum_k h_{k,j}}$$

The global weight uses the document frequency df_i of a term i, which defines the number of documents the term exists. It is irrelevant for the document frequency how often a term occurs in a single document as calculated in term frequency tf. The formula for the df is given according to [Jon04]:

$$df_i = \frac{|d_k : t_i \in d_k|}{|D|}$$

In information retrieval usually the inverse document frequency idf is calculated, due to the semantics that terms that occur in all documents should not have a specific meaning for a single document:

$$idf_i = \log \frac{|D|}{df_i}$$

Consisting of the local term frequency tf and the global (inverse) document frequency idf the TF-IDF weights are created:

$$(tfidf)_{i,j} = \frac{h_{i,j}}{\sum_k h_{k,j}} \times \log \frac{|D|}{df_i}$$

2.2.2 Explicit Semantic Analysis

Explicit Semantic Analysis (ESA) is a relative new method from 2007, that was originally developed for applying it to the wikipedia encyclopedia for finding semantic similarity

between documents [GM07]. The idea behind this method is that a wikipedia article in principle only exists around one topic, a so called concept. As lots of concepts exists in the huge amount of documents of wikipedia, the TF-IDF weights are used for all terms of an article. This results in the term-document matrix, where the columns represent an wikipedia article and the rows represent the terms. The semantic meaning, or concept, correlates to a complete row in the term-document matrix, where the TF-IDF is applied to calculate its relevance. After that, all center vectors of the terms are calculated to know the concepts related to one specific term. With these document vectors a comparison is possible by applying for example the cosine similarity [TP07]. Overall the function of ESA makes it possible to find for every given term and/or document a representation, that considers the relevance of the term or document to all existing concepts.

3 MIRA Framework

In this section we present our graphical user interface (GUI-) application MIRA (Music Information Retrieval Application) that measures similiarity between musical works by applying different IR-methods on chroma features. Specifically in this section we focus on the novel ESA evaluation MIRA is handling as first framework in this area. We start with how we create the audio words in MIRA:

3.1 Creating Audio words from music

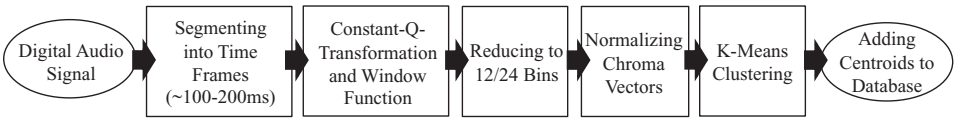


Figure 2: Workflow for creating audio words.

The first step for evaluating musical similarity is creating the needed audio words from the files. The workflow for this implemented in MIRA is outlined in Figure 2. As segmentation, normalization and k-means clustering works straightforward we want to explain the used constant q transform and window function in a bit more detail:

Efficient Constant Q Transform MIRA uses a combined implementation of the Constant Q Transform by Blankertz [Bla01] combined with a Java Implementation from Karl Helgason from 2006. Blankertz optimizes the Constant Q Transform in his algorithm by efficiently multiplying the temporal kernel matrix T into spectral space as suggested in BLA01. This is done by applying FFT on the columns of T. So instead of calculating with $\vec{c} = \vec{x} \cdot T^*$ we calculate efficiently $\vec{c} = \frac{1}{N} x^{ft} \cdot S^*$. Thus S^* is a sparse Matrix with lots of zero values we need much less matrix multiplications than before.

Efficient Window Function As suggested by Blankertz and Brown [Bla01] MIRA uses a Hamming window for balancing frequency errors that happen by randomly chosing time fragments of songs. Such a window function with $\alpha = 0.54,\ \beta = 1 - \alpha = 0.46$. is given by:

$$w(n) = \alpha - \beta\ cos\left(\frac{2\pi n}{N-1}\right)$$

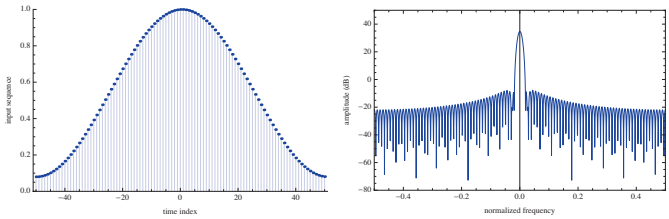


Figure 3: The Hamming window (left) is smoothening the signal. Right we see an application of this window to a signal in the fourier transform.

Figure 3 shows a signal sent without window function - a so called rectangular window on the left and a Hamming window on the right. Clearly, the Hamming window strongly reduces bins with high amplitude, which allows it easier to recognize single frequencies.

3.2 Evaluation of Audio Words

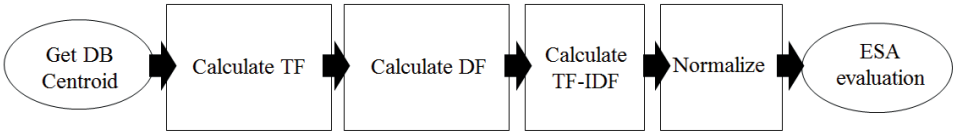


Figure 4: Workflow for evaluating audio words.

As the workflow of MIRA for evaluating audio words is quite intuitive, some considerations need to be done before applying the given IR methods to music:

Applying TF-IDF weights to music As TF-IDF is a method that originally tries to semantically analyze texts, the transfer to music is not easily possible. First note that through clustering created audio words in general have no semantic meaning on their own. Only through TF-IDF a semantic information can be given to every audio word because highly redundant audio words like repeating musical elements are getting a smaller relevance overall. Besides, chromatic information gives no further information on genre or

interpret of a given song. How exactly can we measure the results then? TF-IDF on audio words is useful for recognizing the correct cover versions.

Applying ESA to music For applying ESA to music we analogously to Section 2.2.2 characterize every audio regarding its relevance compared to all given songs of the database. This relevance is calculated by the TF-IDF as described in Paragraph 3.2 and depending on the center of all audio word vectors. As ESA is a method used for very large data sets, our data base of around 4000 songs could be a drawback.

4 Experiments

The experiments done for this paper are two-fold: First, we want to show the quality of the extracted chroma features for finding similarity between cover versions of a song to the original. Second, we apply and evaluate ESA on our database for optimizing the ranking result of similar audio files.

4.1 The quality of chroma features

Lets take a look how effective different chroma features can be differentiated for different interpretations and tonalities in the audio files:

Classical Music Example The highest quality and the highest musical similarity with chroma features on different interpreted audio files can be achieved with classical music. Here in Figure 5 can be even visually, with human eye, shown that chroma features give a solid basis for musical similarity.

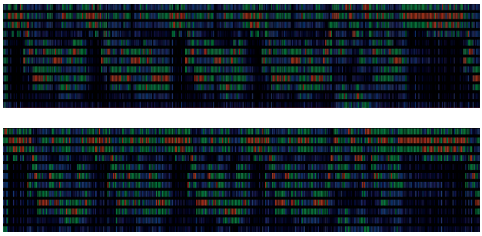


Figure 5: Comparison: Ludwig van Beethoven (original), title „For Elise“, interpreted by Sylvia Capova (upper) und Ivo Pogorelich (lower)

Same tonality but polyphonic voices Although given the same tonality as in the classical music example but with quite different stylistic cover especially with many polyphonic

voices, the song has very little similarities as can be seen in Figure 6.

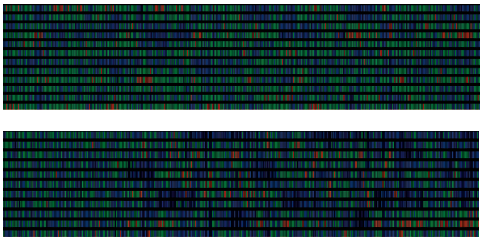


Figure 6: Comparison: Lykke Li „I Follow Rivers“, interpreted by Lykke Li (upper) and Triggerfinger (lower)

Different tonalities The last comparison shows two songs, that are stylistically similar but interpreted in different tonalities. Figure 7 shows that the chromagram does indeed show similar pattern, but the displacement of the chroma vector components makes a mapping of identical audio words without correction impossible.

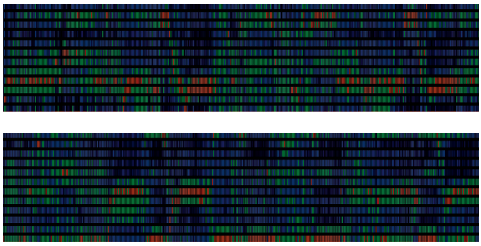


Figure 7: Comparison: Hank Williams „Ramblin’ Man“, interpreted by Hank Williams (upper) and Del Shannon (lower)

4.1.1 Explicit Semantic Analysis Evaluation

The following evaluation refers to a given database with around 4000 songs, as described in a specification by Riley, Heinen and Ghosh [RHG08] as well as Constant-Q-parameter settings described by Brown [BP92]. By applying a standard evaluation within a feature space applying the cosine similarity as done in the above mentioned paper, some cover versions are not recognized accurately. To improve the ranking we newly want to test the explicit semantic analysis (ESA). As ESA has high computational costs we only apply it to already falsely ranked results.

Ranking with ESA For testing the ranking of ESA we chose two songs, which were as live covers of „Wicked Game“ from Chris Isaak as well as „Bad Moon Rising“ from Creedence Clearwater Revival falsely ranked compared to their studio version.

Rang	Song 1	Song 2	Winkel
1	*Wicked Game* Chris Isaak	*Wicked Game* Chris Isaak (live)	62,0°
21	*Wicked Game* Chris Isaak	*Wicked Game* Chris Isaak (live)	46,9°

Figure 8: Comparison of the Chris Isaak song „Wicked Game“. Red: Bad ranking without ESA. Green: Correct ranking with ESA. ”Winkel” refers to the cosine similarity

Rang	Song 1	Song 2	Winkel
7	*Bad Moon Rising* Creedence Clearwater Revival	*Bad Moon Rising* Creedence Clearwater Revival (live)	71,3°
26	*Bad Moon Rising* Creedence Clearwater Revival	*Bad Moon Rising* Creedence Clearwater Revival (live)	45,3°

Figure 9: Comparison of the Creedence Clearwater Revival song „Bad Moon Rising“. Red: Bad ranking without ESA. Green: Correct ranking with ESA. ”Winkel” refers to the cosine similarity

Our experiment shows for both song examples a much more effective recognition of the cover-versions when ESA is applied. While the cover identification of „Bad Moon Rising“ by Creedence Clearwater Revival was raised from rank 26 to rank 7, has „Wicked Game“ by Chris Isaak even improved to rank 1 - 20 ranks higher than before.

Scalability of ESA Clearly, the ESA evaluation proved useful for improving false ranks, but still the roots of ESA has to be considered: ESA is a text-retrieval method and as such conceptionally prone to work on a much larger amount of terms and documents as is the case here. Although Riley, Heinen and Ghosh claim a number of 500 audio words as the most effective [RHG08] without any proof, we experiment by scaling the number of audio words for ESA. As we have no larger database of songs than 4000 we only scale the number of audio words, especially the k-means cluster centers, from 500 to 10000. By doing so we receive no further positive effect on the ranking against our hypothesis for further improvement.

5 Conclusion

Concluding this paper we showed first, how chroma features are interesting and reliable for differentiating cover versions of music files. Furthermore we exploited how different IR methods like TF-IDF weights and explicit semantic analysis (ESA) are useful for optimizing the ranking in musical similarity analysis. We determined this by vast experiments done with our GUI - application MIRA. We see future work in analyzing different kind of features than chroma vectors to efficiently answer other research questions than how close cover versions are related to the original.

References

[Bla01] Benjamin Blankertz. The constant Q transform, 2001.

[BP92] Judith C. Brown and Miller S Puckette. An efficient algorithm for the calculation of a constant Q transform. *The Journal of the Acoustical Society of America*, 92:2698, 1992.

[Bro90] Judith C. Brown. *Calculation of a constant Q spectral transform*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1990.

[BYRN11] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, London, 2011.

[Dow09] J Stephen Downie. Music Information Retrieval Meets Education. *The Music Information Retrieval Evaluation Exchange MIREX*, pages 247–255, 2009.

[GM07] Evgeniy Gabrilovich and Shaul Markovitch. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

[GMS12] Peter Grosche, Meinard Mller, and Joan Serra. Audio Content–Based Music Retrieval. *Multimodal Music Processing*, pages 157–174, 2012.

[Jon04] Karen Sp rck Jones. IDF term weighting and IR research lessons. *Journal of Documentation*, 60(5):521–523, 2004.

[Lug09] Michael Lugmair. *Ein Dynamic Time Warping basiertes Score Following System*. PhD thesis, Universitt Augsburg, 2009.

[RHG08] Matthew Riley, Eric Heinen, and Joydeep Ghosh. A Text Retrieval Approach to Content-Based Audio Retrieval. *Ismir 2008, Session 3a, Content-Based Retrieval, Categorization and Similarity I*, pages 295–300, 2008.

[SY73] Gerard Salton and Chung-Shu Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372, 1973.

[TP07] Sandeep Tata and Jignesh M Patel. Estimating the selectivity of tf-idf based cosine similarity predicates. *ACM SIGMOD Record*, 36(2):7–12, 2007.

Vergleich von Distanzen und Kernel für Klassifikatoren zur Optimierung der Annotation von Bildern

Alexander Askinadze

Heinrich-Heine-Universität Düsseldorf
Universitätsstrasse 1
D-40225 Düsseldorf
alexander.askinadze@uni-duesseldorf.de

Abstract: Die stetig steigende Anzahl von Bildern erfordert Verfahren zur maschinellen Annotation. Um automatisch semantische Informationen aus den Bildern zu extrahieren, repräsentieren wir die Bilder durch numerische Vektoren, sogenannte BoW-Histogramme und klassifizieren diese auf vorgegebene Klassen. Als Klassifikatoren werden Nearest-Centroid (NC) und Support Vector Machine (SVM) eingesetzt. Auf der Caltech 101 Bilder-Datenbank liefert der SVM-Klassifikator mit dem empfohlenen RBF-Kernel bessere Ergebnisse als der NC-Klassifikator mit der Euklidischen Distanz. Wir vergleichen verschiedene Distanzfunktionen wie z.B. die Bhattacharyya- und Hellinger-Distanz und zeigen, wie sich die Mahalanobis-Distanz für eine Modifikation des NC-Klassifikators nutzen lässt. Nach einer Evaluation folgern wir, dass der NC-Klassifikator mit anderen Distanzfunktionen die SVM-Ergebnisse erreichen kann und eine Normierung der BoW-Histogramme sich ebenfalls positiv auswirkt. Außerdem zeigen wir, dass sich die Ergebnisse des SVM-Klassifikators signifikant durch den Einsatz des Chi-Quadrat- und Histogrammschnitt-Kernels verbessern können.

1 Einleitung

Die stetig wachsende Anzahl von Bildern erfordert einen erheblichen Arbeitsaufwand durch manuelle Annotation. Die Annotation ist notwendig, um Bilder innerhalb der großen Bestände wiederfinden zu können. Maschinell ausgeführte Annotationsverfahren werden dadurch immer wichtiger. Diese Verfahren müssen in der Lage sein, den semantischen Inhalt der Bilder zu erkennen und dadurch passende Metainformationen zu extrahieren.

Ein typischer Ansatz für diese Aufgabe ist die Umwandlung eines Bildes in einen numerischen Vektor und die anschließende Klassifikation dieses Vektors auf n vorgegebene Klassen, die sich als Annotationen des Bildes verwenden lassen. Die Qualität der Annotation eines Bildes hängt also von der Qualität der numerischen Repräsentation des Bildes und des Klassifikators ab.

Nach der Wahl einer festen numerischen Repräsentation des Bildes soll untersucht werden, welche Faktoren und Parametereinstellungen beliebter Klassifikatoren sich positiv auf die Klassifikationsergebnisse auswirken können. Hierzu werden die Klassifikatoren mit üblichen oder empfohlenen Einstellungen getestet und anschließend wird gezeigt, wie

sich die Wahl anderer Einstellungen auswirkt.

2 Grundlagen

Ein beliebter Ansatz zur Transformation eines Bildes in einen numerischen Vektor ist das Prinzip “bag of keypoints” [CDF+04], dass auf das “Bag of Words (BoW)”-Modell zurückzuführen ist. Die Bilder werden hierbei in eine einheitliche Darstellung von gleichdimensionalen Histogrammen (BoW-Histogrammen) gebracht.

Ablauf der Tansformation Mit einem Detektor werden markante Punkte im Bild gefunden und durch Deskriptoren beschrieben. Zur Detektion und Extraktion der Bildmerkmale wird das SIFT (Scalable Invariant Feature Transform)-Verfahren [Low04] verwendet, dass nach [vdSGS08] vergleichsweise gute deskriptive Eigenschaften aufweist. Das SIFT-Verfahren beschreibt die gefundenen markanten Punkte durch 128-dimensionale Deskriptoren. Die Anzahl der SIFT-Deskriptoren aller Trainingsbilder wird durch Clustering (K-Means) auf K, durch das Clustering entstandene Zentroide reduziert. Basierend auf diesem Clusteringergebnis der SIFT-Deskriptoren aller Trainingsbilder können die K-dimensionalen BoW-Histogramme für die Bilder erstellt werden. Zu beachten ist, dass die Dimension K der BoW-Histogramme sich direkt aus der Vorauswahl des Parameters K im K-Means Algorithmus ergibt.

Klassifikation Sobald ein Bild durch ein BoW-Histogramm repräsentiert ist, kann dieses als Eingabe an einen Klassifikator weitergegeben werden. Als Klassifikatoren lassen sich beispielsweise die beliebten Klassifikatoren Nearest-Centroid (NC) auch als ”Rocchio classification” [MRS08] bekannt, Abwandlungen von NC: ”Center-based nearest neighbor classifier”(CNN) [GW07] oder Support Vector Machine (SVM) [CV95] einsetzen. Der Klassifikationsvorgang für ein neues Bild läuft wie in Abbildung 1 ab.

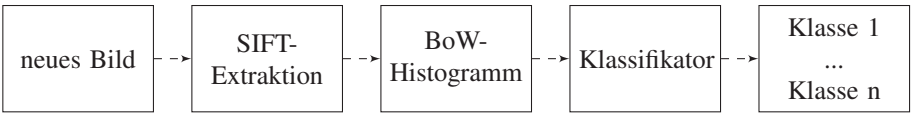


Abbildung 1: Klassifikationsablauf für ein neues Bild

Baseline Im weiteren Verlauf des Papers soll untersucht werden, welche Parametereinstellungen in einzelnen Schritten des Klassifikationsvorgangs positive Auswirkungen auf das Klassifikationsergebnis haben. Hierzu wird die Bild-Datenbank Caltech 101 [FFFP04] mit über 9000 Bildern aus 102 Klassen verwendet. Diese Bilddatenbank eignet sich zur Evaluation, da sie aus vielen semantisch unterschiedlichen Klassen (Flugzeuge, Insekten, Pflanzen, ...) besteht und sich somit nicht auf einen speziellen semantischen Bereich

spezialisiert. Als Baseline sollen die Klassifikatoren NC und SVM mit üblichen Parametereinstellungen dienen, deren Ergebnisse in Tabelle 1 abgebildet sind. Pro Klasse werden hierbei je 15 Trainings- und Testbilder verwendet. Jeder Versuch wird 3 mal wiederholt. Für den NC-Klassifikator wird wie in [MRS08] die Euklidische Distanz verwendet. Für die SVM werden die in [wHcCjL10] empfohlenen Einstellungen zur Skalierung der Vektoren und zur Wahl des SVM-Kernels (RBF-Kernel) verwendet. Für die Multiklassen-SVM-Strategie wird die in [CL11] empfohlene "one-against-one" - Strategie verwendet.

	100		500		1000		1500	
	Pr	Re	Pr	Re	Pr	Re	Pr	Re
NC (Euklidische Dist.)	0,25	0,25	0,31	0,29	0,36	0,33	0,38	0,32
SVM (RBF-Kernel)	0,3	0,3	0,38	0,35	0,39	0,35	0,41	0,36

Tabelle 1: Precision- und Recallergebnisse mit Standardeinstellungen für die BoW-Histogramm-Größen 100,500,1000 und 1500.

2.1 Distanzfunktionen

Es folgt eine Übersicht über einige Klassen von Distanzfunktionen.

2.1.1 Durch Normen induzierte Distanzfunktionen

Eine Norm ist eine Abbildung $||.|| : V \rightarrow \mathbb{R}_+, x \rightarrow ||x||$ von einem Vektorraum über einem Körper (in unserem Fall $\mathbb{R}$) in die Menge $\mathbb{R}_+ = \{x \in \mathbb{R} | x \geq 0\}$. Eine ganze Reihe von Normen, die p-Normen lassen sich durch $||x||_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$ mit $x \in \mathbb{R}^n$ und $p \in \mathbb{R}, 1 \leq p \leq \infty$ angeben.

Bekannte p-Normen sind die 1-Norm (induziert die Manhattan Distanz), 2-Norm (induziert die Euklidische Distanz) und für $p \rightarrow \infty$ die Maximum Distanz. Mit Hilfe der folgenden Gleichung lassen sich aus beliebigen p-Normen Distanzen ableiten: $d(x, y) = ||x - y||$.

2.1.2 Durch Matrizen induzierte Distanzfunktionen

Eine weitere Klasse von Distanzfunktionen lässt sich durch eine Matrix A angeben:

$$d_A(x, y) = ||x - y||_A = \sqrt{(x - y)^T A (x - y)} \tag{1}$$

Ist A die Einheitsmatrix, so entspricht d_A der Euklidischen Distanz. Interessant ist also, wie sich von der Einheitsmatrix abweichende Matrizen auf das Klassifikationsergebnis auswirken und wie diese Matrizen möglichst passend zu den Daten bestimmt werden können.

Mahalanobis-Distanz Eine Möglichkeit solche Matrizen zu bestimmen, bietet sich aus der Berechnung der Kovarianzmatrix S und der Nutzung ihrer inversen Matrix S^{-1}. Die Distanzfunktion $d_{S^{-1}}$ heißt Mahalanobis-Distanz [MT09].

Es ist zunächst unklar, wie eine Kovarianzmatrix für eine Trainingsmenge mit n p-dimensionalen BoW-Histogrammen zu berechnen ist. Wir benutzen einen statistischen Ansatz zur Schätzung einer Kovarianzmatrix aus Daten [Hel09] und wenden diesen Ansatz folgendermaßen auf die vorliegenden Trainings-BoW-Histogramme an: Sei $X = (X_1, ..., X_p)$ ein multivariater Zufallsvektor der Zufallsvariablen $X_i, 1 \leq i \leq p$. Wir nehmen an, dass die BoW-Histogramme durch X realisiert wurden. Die Kovarianz von X lässt sich berechnen mit:

$$Cov(X) = \begin{pmatrix} Cov(X_1, X_2) & Cov(X_1, X_2) & ... & Cov(X_1, X_p) \\ Cov(X_2, X_1) & Cov(X_2, X_2) & ... & Cov(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_p, X_1) & Cov(X_n, X_2) & ... & Cov(X_p, X_p) \end{pmatrix} \tag{2}$$

Die multivariaten Verteilungsparameter der X_i liegen nicht vor und müssen deshalb geschätzt werden. Durch die vorliegenden Trainings-BoW-Histogramme liegen die $p-dimensionalen$ Realisationen von X vor, aus denen geschätzt werden kann. Sei $B = (b_{ij})$ eine $n \times p$ Matrix mit den BoW-Trainings-Histogrammen in den Zeilen. Seien $\overline{b_j}$ die Mittelwerte der Spalten. Dann gewinnen wir die zentrierte Matrix $B^* = (b^*_{ij})$ durch $b^*_{ij} = b_{ij} - \overline{b_j}$. Die Kovarianz $Cov(X_i, X_j)$ können wir dann mit der korrigierten Stichprobenkovarianz $Cov(X_j, X_k) = \frac{1}{n-1} \sum_{i=1}^{n} b^*_{ij} b^*_{ik}$ schätzen und die $p \times p$ Kovarianzmatrix $S = Cov(X) = (Cov(X_i, X_j))$ ist damit komplett angegeben.

Aus der Annahme, dass die Trainings-Histogramme aus der gleichen Klasse durch die gleiche Verteilung entstanden sind, berechnen wir pro Klasse eine solche Kovarianzmatrix. Damit ergibt sich pro Klasse eine Mahalanobis-Distanzfunktion, mit der sich der Abstand eines Bildes zu einer Klasse mit

$$d_C(x) = \sqrt{(x - \mu_C)^T S_C^{-1} (x - \mu_C)} \tag{3}$$

berechnen lässt, wobei μ_C der Mittelwert über die Trainings-BoW-Histogramme der Klasse C und S_C die Kovarianzmatrix der Trainingsvektoren der Klasse C ist.

2.1.3 Bhattacharyya-Distanz

Aus dem Bhattacharyya-Koeffizienten $BC(x, y) = \sum_{i=1}^{n} \sqrt{x_i y_i}$ (Ähnlichkeitsfunktion) für zwei BoW-Histogramme $x, y \in \mathbb{R}$ lässt sich die Bhattacharyya-Distanz mit

$$d_{B_1}(x, y) = -ln(BC(x, y)) \tag{4}$$

angeben [Cha07]. Falls $0 \leq BC(x, y) \leq 1$, dann ist $0 \leq d_{B_1} \leq \infty$.

2.1.4 Hellinger-Distanz

Eine weitere Möglichkeit aus dem Bhattacharyya-Koeffizienten $BC(x, y)$ (siehe 2.1.3) eine Distanz abzuleiten [Cha07] ist ebenfalls für $0 \leq BC(x, y) \leq 1$ gegeben mit:

$$d_{B_2}(x, y) = \sqrt{1 - BC(x, y)} \in [0, 1] \tag{5}$$

2.1.5 Histogrammschnitt-Distanz

Ähnlich wie bei der Bhattacharyya-Distanz gewinnen wir hier aus einem Ähnlichkeitsmaß eine Distanzfunktion. Aus dem Histogrammschnitt $\sum_{i=1}^{n} min(x_i, y_i) \in [0, 1]$ für zwei normierte BoW-Histogramme $x, y \in \mathbb{R}^n$ leiten wir diese Distanz ab:

$$d_H(x, y) = 1 - \sum_{i=1}^{n} min(x_i, y_i) \in [0, 1] \tag{6}$$

2.2 Ziele

Zur Verbesserung der Klassifikationsgüte wollen wir folgende Parameter untersuchen:

- Distanzfunktionen: Viele Klassifikatoren arbeiten basierend auf Distanzen, indem sie die Distanzen zwischen den BoW-Histogrammen berechnen. In der Literatur wird oft die Euklidische Distanz verwendet. Nun soll die Auswirkung weiterer Distanzfunktionen auf das Ergebnis untersucht werden.
- Kernel: Für den SVM-Klassifikator wird in der Anleitung der SVM-Bibliothek LibSVM [CL11] der RBF-Kernel empfohlen, da er allgemein gute Ergebnisse liefern soll. Nun soll die Klassifikationsgüte von SVM mit anderen Kernel untersucht werden.
- Normierung der BoW-Histogramme: Es soll der Einfluss der Normierung auf die Klassifikationsgüte der Distanz-basierten Klassifikatoren untersucht werden.

3 Hauptteil: Optimierung der Klassifikatoren

3.1 Optimierung NC-basierter Klassifikatoren

NC-basierte Klassifikatoren ordnen einer Eingabe x die Klassen C nach dem Prinzip $\arg\min_C d_C(x)$ zu, wobei $d_C(x)$ der Abstand von x zum Zentrum der Klasse C ist. Üblicherweise wird hier die Euklidische Distanz verwendet. Wir untersuchen nun, welche Auswirkungen weitere Distanzen und modifizierte NC-Ansätze auf die Ergebnisse haben.

3.1.1 NC-Klassifikator

NC-Klassifikator mit normierten BoW-Histogrammen

	100		250		500		1000		1500	
	Pr	Re	Pr	Re	Pr	Re	Pr	Re	Pr	Re
(1*)	0,27	0,27	0,31	0,31	0,34	0,32	0,33	0,32	0,34	0,32
(2*)	0,27	0,26	0,32	0,29	0,32	0,29	0,38	0,32	0,39	0,32
(3*)	0,27	0,26	0,31	0,3	0,31	0,3	0,35	0,34	0,36	0,33
(4*)	0,28	0,28	0,3	0,31	0,3	0,31	0,39	0,36	0,39	0,35
(5*)	0,28	0,27	0,34	0,32	0,33	0,31	0,38	0,35	0,41	0,36

Tabelle 2: Klassifikationsergebnisse des NC-Klassifikators mit normierten BoW-Histogrammen für die BoW-Größen 100,250,500,1000 und 1500 sowie den Distanzfunkionen: (1*) Manhatten (2*) Euklid (3*) HistoIntersection (4*) Bhattacharyya (5*) Hellinger

NC-Klassifikator mit nicht normierten BoW-Histogrammen

	100		250		500		1000		1500	
	Pr	Re	Pr	Re	Pr	Re	Pr	Re	Pr	Re
(1*)	0,24	0,21	0,26	0,2	0,25	0,15	0,19	0,08	0,13	0,07
(2*)	0,24	0,23	0,29	0,27	0,3	0,29	0,34	0,3	0,34	0,3

Tabelle 3: Klassifikationsergebnisse des NC-Klassifikators mit nicht normierten BoW-Histogrammen für die BoW-Größen 100,250,500,1000 und 1500 sowie den Distanzfunkionen: (1*) Manhatten (2*) Euklid

Diskussion zum NC-Klassifikator Mit normierten BoW-Histogrammen erreicht der NC-Klassifikator mit der BoW-Größe 1500 sowie der Hellinger-Distanz das Ergebnis (Recall: 0.36, Precision: 0.41), womit er die SVM-Ergebnisse der Baseline einholt und die NC-Ergebnisse der Baseline deutlich übertrifft. Das zeigt, dass auch einfach zu implementierende Klassifiktoren sich zur weiteren Verwendung und Optimierung eignen. Insbesondere sind NC-Klassifikatoren interessant, weil sie schnellere Trainingszeiten haben, als SVM-Klassifikatoren. Für nicht normierte Histogramme können wir nur die normierungsunabhängigen Distanzen vergleichen. Hier sehen wir, dass die Ergebnisse mit nicht normierten BoW-Histogrammen schlechter sind. Für die Manhatten-Distanz zeigt sich hier ein signifikanter Unterschied.

3.1.2 CNN

CNN (Center-based nearest neighbour) [GW07] ist ein modifizierter NC-Ansatz und berechnet nicht den Abstand eines BoW-Histogramms x zum Klassenzentrum o^c, sondern den Abstand von x zu seiner Projektion auf der Strecke $\overline{x_i^c o^c}$, wobei x_i^c ein Punkt der

Klasse c ist. Da es N_c Trainingspunkte in der Klasse c gibt, werden N_c Projektionen berechnet und anschließend minimiert, sodass die Distanz zur Klasse c sich mit

$$d^c_{CNN} = \min_{1 \leq i \leq N_c} d(x, \overline{x^c_i o^c}) \tag{7}$$

berechnen lässt. Die Ergebnisklasse wird nochmals aus allen Klassendistanzen minimiert: $c_{Result} = \arg\min_{c=1,2,\ldots C} d^c_{CNN}$. Im Originalpaper wird die Euklidische Distanz verwendet. Wir untersuchen hier den Einsatz weiterer Distanzen.

CNN mit normierten BoW-Histogrammen

	100		250		500		1000	
	Pr	Re	Pr	Re	Pr	Re	Pr	Re
(1*)	0,28	0,26	0,34	0,32	0,33	0,34	0,38	0,35
(2*)	0,28	0,27	0,34	0,3	0,36	0,33	0,4	0,34
(3*)	0,31	0,29	0,36	0,33	0,35	0,35	0,37	0,34

Tabelle 4: Ergebnisse des CNN-Klassifikators mit normierten BoW-Histogrammen für die BoW-Größen 100,250,500 und 1000 sowie den Distanzfunktionen: (1*) Manhatten (2*) Euklid (3*) HistoIntersection

CNN mit nicht normierten BoW-Histogrammen

	100		250		500		1000	
	Pr	Re	Pr	Re	Pr	Re	Pr	Re
(1*)	0,28	0,26	0,31	0,25	0,29	0,19	0,26	0,1
(2*)	0,29	0,28	0,33	0,31	0,35	0,33	0,36	0,33

Tabelle 5: Ergebnisse des CNN-Klassifikators mit nicht normierten BoW-Histogrammen für die BoW-Größen 100,250,500 und 1000 sowie den Distanzfunktionen: (1*) Manhatten (2*) Euklid

Diskussion des CNN-Klassifikators Der CNN-Klassifikator liefert für normierte BoW-Histogramme mit der BoW-Größe von $k = 1000$ die besten Ergebnisse. Hierbei liefert die Manhatten Distanz einen Recall-Wert von 0.35 und kommt damit nah an den besten Recall-Wert (0.36) des SVM-Klassifikators aus der Baseline heran. Bei nicht normierten BoW-Histogrammen sind die Ergebnisse durchgehend schlechter, wobei auch hier der deutliche Unterschied bei der Manhattan Distanz zu bemerken ist.

3.1.3 Mahalanobis-Distanz-Klassifikator

Dieser Klassifikator ergibt sich aus der Mahalanobis-Distanz (3). Die beste Klasse für ein durch das BoW-Histogramm x repräsentiertes Bild lässt sich dann als eine Modifikation des NC-Klassifikators mit $\arg\min_C d_C(x)$ berechnen.

Die Ergebnisse des Mahalanobis-Distanz-Klassifikators sind in Tabelle 6 abgebildet.

	100		250		500		1000	
	Pr	Re	Pr	Re	Pr	Re	Pr	Re
(1*)	0,2	0,22	0,27	0,27	0,31	0,3	0,38	0,32
(2*)	0,3	0,28	0,34	0,32	0,34	0,33	0,39	0,37

Tabelle 6: Klassifikationsergebnisse des Mahalanobis-Distanz-Klassifikators für die BoW-Größen 100,250,500 und 1000 sowie für folgende Einstellungen: (1*) normierte BoW-Histogramme (2*) nicht normierte BoW-Histogramme

Diskussion für den Mahalanobis-Distanz-Klassifikator Bereits ab der BoW-Histogramm-Größe von 1000 und nicht normierten Histogrammen übertrifft der NC-Klassifikator mit der Mahalanobis-Distanz die Ergebnisse des NC-Klassifikators mit der Euklidischen Distanz und das beste Recall-Ergebnis des SVM-Klassifikators mit dem RBF-Kernel. Anders als bei vorherigen Versuchen wirkt sich die Normierung deutlich negativ aus.

3.2 Optimierung des SVM-Klassifikators

3.2.1 SVM-Kernel

Folgende Kernel werden zur Klassifikation der BoW-Histogramme untersucht:

- RBF-Kernel (Radial Basis Function): $K(u,v) = exp(-\frac{1}{2\sigma^2}||u-v||^2)$. Der σ-Parameter wird mit der folgenden Heuristik [JDH99] berechnet:
  $\sigma = median(\{dist(u,v)|u \neq v\})$.
- Polynom-Kernel: $K(x,y) = (\alpha x^T y + c)^d$
- Linear-Kernel (Spezialfall des Polynom-Kernels): $K(x,y) = x^T y + c$
- Hyperbolic Tangent (Sigmoid) Kernel: $K(x,y) = tanh(\alpha x^T y + c)$
- Inverse Multiquadric Kernel [Mic86]: $K(x,y) = \frac{1}{\sqrt{||x-y||^2+1}}$
- χ^2 Kernel: $K(x,y) = 1 - \sum_{i=1}^{n} \frac{(x_i-y_i)^2}{\frac{1}{2}(x_i+y_i)}$, abgeleitet aus [Cha07] und nur für normierte BoW-Histogramme sinnvoll
- Histogrammschnitt-Kernel: $K(x,y) = \sum_{i=1}^{n} min(x_i, y_i)$

3.2.2 Evaluation verschiedener Kernel

Für die Evaluation wird die DAG Multiklassen-Strategie verwendet. Bei DAG (Directed Acyclic Graph) [PCST99] handelt es sich um einen azyklisch gerichteten Graphen. In jedem Knoten des Graphen wird eine binäre Entscheidung über die Klassenzugehörigkeit eines betrachteten BoW-Histogramms getroffen. In Tabelle 7 wird nur das Ergebnis für $C = 2$ präsentiert, wobei C der C-SVM-Parameter [CL11] ist.

	100		250		500		1000		1500	
	Pr	Re	Pr	Re	Pr	Re	Pr	Re	Pr	Re
(1*)	0,31	0,29	0,38	0,36	0,4	0,37	0,43	0,4	0,42	0,38
(2*)	0,31	0,28	0,37	0,32	0,35	0,31	0,4	0,32	0,4	0,31
(3*)	0,32	0,3	0,38	0,36	0,4	0,37	0,41	0,39	0,41	0,38
(4*)	0,25	0,2	0,32	0,24	0,29	0,21	0,33	0,22	0,32	0,2
(5*)	0,26	0,2	0,31	0,23	0,31	0,22	0,35	0,23	0,29	0,21
(6*)	0,26	0,2	0,32	0,22	0,31	0,21	0,32	0,21	0,33	0,21
(7*)	0,27	0,21	0,3	0,23	0,31	0,23	0,36	0,23	0,32	0,21

Tabelle 7: Klassifikationsergebnisse für die BoW-Größen 100,250,500,1000 und 1500 sowie folgende Kernel: (1*) χ^2 (2*) RBF mit σ-Heuristik (3*) Histogramm-Schnitt (4*) Inverse Multiquadric (5*) Linear ($c = 1$) (6*) Polynomial ($\alpha = 1, c = 1, d = 3$) (7*) Sigmoid ($c = 1$)

Diskussion der verschiedenen SVM-Kernel Die Kernel χ^2- und der Histogrammschnitt-Kernel verbessern die SVM-Baseline Ergebnisse deutlich. So wird mit dem χ^2-Kernel der Recall-Wert von 0.36 auf 0.4 und der Precision-Wert von 0.41 auf 0.43 verbessert. Zu bemerken ist, dass die RBF-Kernel σ-Heuristik für schlechtere Recall-Werte sorgt, die Precision-Werte aber nicht deutlich schlechter werden. Die polynomiellen Kernel müssen noch weiter untersucht werden, da ein 3-dimensionales Beispiel nicht zur Ablehnung ausreicht. Der Inverse Multiquadric und Sigmoid Kernel sind aufgrund vergleichsweise schlechter Recall-Werte nicht für diese Aufgabe empfehlenswert.

4 Fazit

Die distanzbasierte Optimierung hat gezeigt, dass ein vergleichsweise einfacher Klassifikator, wie der NC-Klassifikator mit der Hellinger-Distanz ähnlich gute Ergebnisse wie eine SVM mit dem oft empfohlenen RBF-Kernel liefern kann. Auch der Mahalanobis-Distanz Klassifikator konnte das Baseline Ergebnis des NC-Klassifikators deutlich verbessern. Dies zeigt, dass ein Vergleich von Distanzen sinnvoll ist, da das Ergebnis so erheblich beinflusst werden kann. Andererseits stellt sich die Frage, ob die Wahl der Distanzfunktion allein aus der Definition und nicht aus Experimenten bestimmt werden kann. So bleibt zu prüfen, ob die Hellinger Distanz auch auf normierten Histogrammen aus anderen Bereichen gute Ergebnisse liefert, um eine allgemeine Empfehlung für solche Daten auszusprechen. Da NC-Klassifikatoren einfach zu implementieren sind und im Vergleich zu SVM-Klassifikatoren schnellere Trainingszeiten aufweisen, können die vergleichsweise guten Ergebnisse als Motivation zur weiteren Untersuchung und Optimierung dienen.

Nicht zu vernachlässigen ist auch der Einfluss der Normierung auf die Klassifikationsergebnisse. In den meisten Fällen hat die Normierung positive Tendenzen gezeigt, außer bei der Mahalanobis-Distanz. Insbesondere sollte die Normierung für den Einsatz der Manhatten Distanz berücksichtigt werden, da hier die größten Differenzen aufgetreten sind.

Ein weiterer zu berücksichtigender Faktor ist die Wahl der Kernels für die SVM. Auch hier

kann die Wahl eines anderen geeigneteren Kernels deutliche Verbesserungen hervorrufen. Die Kernel χ^2- und Histogrammschnitt-Kernel können für die Klassifikation der BoW-Histogramme aufgrund der eindeutig besseren Ergebnisse empfohlen werden.

Literatur

[CDF+04] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski und Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, Seiten 1–22, 2004.

[Cha07] Sung-Hyuk Cha. Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307, 2007.

[CL11] Chih-Chung Chang und Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[CV95] Corinna Cortes und Vladimir Vapnik. Support-Vector Networks. In *Machine Learning*, Seiten 273–297, 1995.

[FFFP04] L. Fei-Fei, R. Fergus und Pietro Perona. Learning Generative Visual Models From Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. 2004.

[GW07] Qing-Bin Gao und Zheng-Zhi Wang. Center-based nearest neighbor classifier. *Pattern Recognition*, 40(1):346–349, 2007.

[Hel09] Reiner Hellbrück. Multivariate Korrelationsanalyse. In *Angewandte Statistik mit R*, Seiten 165–188. Springer, 2009.

[JDH99] Tommi Jaakkola, Mark Diekhans und David Haussler. Using the Fisher Kernel Method to Detect Remote Protein Homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, Seiten 149–158. AAAI Press, 1999.

[Low04] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[Mic86] Charles A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.

[MRS08] Christopher D. Manning, Prabhakar Raghavan und Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[MT09] P. Mather und B. Tso. *Classification Methods for Remotely Sensed Data, Second Edition*. Environmental engineering. Taylor & Francis, 2009.

[PCST99] John C Platt, Nello Cristianini und John Shawe-Taylor. Large Margin DAGs for Multiclass Classification. In *nips*, Jgg. 12, Seiten 547–553, 1999.

[vdSGS08] K. E. A. van de Sande, T. Gevers und C. G. M. Snoek. Evaluation of Color Descriptors for Object and Scene Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[wHcCjL10] Chih wei Hsu, Chih chung Chang und Chih jen Lin. A practical guide to support vector classification, 2010.

Aspekte einer automatischen Meinungsbildungsanalyse von Online-Diskussionen

Matthias Liebeck

Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
Universitätsstr. 1
40225 Düsseldorf
liebeck@cs.uni-duesseldorf.de

Abstract: Heutzutage haben Menschen die Möglichkeit, ihre Meinung zu verschiedensten Themen in onlinebasierten Diskussionsplattformen zu äußern. Diese Meinungen können in Form einer Meinungsbildungsanalyse genauer untersucht werden. In diesem Beitrag werden verschiedene Aspekte einer automatisierten Diskussionsverfolgung untersucht. Dazu werden Analysekriterien definiert und die vorgestellten Ansätze auf zwei deutschsprachige Datensätze angewendet.

1 Einleitung

Das Internet ermöglicht es Menschen in onlinebasierten Plattformen ihre Meinung über Themen, z.B. politische Ereignisse oder Konsumgüter, in Form von Textbeiträgen, unter anderem in Foren, als Kommentare auf Nachrichtenportalen oder in sozialen Netzwerken, öffentlich preiszugeben. Die über ein Thema gemachten Äußerungen können individuell analysiert werden, um daraus im Rahmen einer Meinungsbildungsanalyse ein Meinungsbild zu erstellen. Aus Sicht einer Firma kann eine automatisierte Untersuchung von Social Media gegebenenfalls Hinweise auf die öffentliche Meinung über firmeneigene Produkte liefern. Im Rahmen eines Online-Partizipationsverfahrens, bei dem sich beispielsweise Bürger bei einer Diskussion über lokalkommunale Entscheidungen beteiligen können, kann die Haltung zu bestimmen Themen untersucht werden.

Durch die große Anzahl an Beiträgen ist eine automatisierte Analyse erstrebenswert. Unter Einsatz von Techniken der maschinellen Sprachverarbeitung können durch eine **Natural Language Processing (NLP) Pipeline** Meinungen in Textform analysiert und für eine weitere Verarbeitung aufbereitet werden. Anwendungsabhängig kann ermittelt werden, über welches Thema eine Aussage gemacht wird und welche **Tonalität** $t \in \{$positiv, neutral, negativ$\}$ dabei ausgedrückt wird. So erfolgt beispielsweise im Satz „*Peter liebt Schokolade.*“ eine positive Aussage über das Thema *Schokolade*. Im Rahmen dieser Arbeit werden Kommentare eines deutschsprachigen Nachrichtenportals und eines Online-Partizipationsverfahrens untersucht und themenspezifische Meinungsbilder in Hinblick auf einen zeitlichen Aspekt konstruiert.

Zur maschinellen Sprachverarbeitung deutschsprachiger Kommentartexte wird eine NLP-Pipeline eingesetzt, die sequenziell mehrere Verarbeitungsschritte durchläuft. Zu Beginn wird OpenNLP[1] zur Trennung eines Eingabetextes in mehrere Sätze, zur Zerlegung der Sätze in einzelne Wörter bzw. **Tokens** und für ein anschließendes POS-Tagging verwendet, bei dem für einzelne Wörter ein **Part-of-Speech-Tag (POS-Tag)** bzw. eine Wortart aus dem *Stuttgart-Tübingen-Tagset (STTS)* [STST99] bestimmt wird.

Um neben POS-Tags Informationen über weitere grammatikalische Eigenschaften aus einem Satz zu erhalten, werden **Dependenzen** bestimmt, die jeweils eine Abhängigkeit zwischen zwei Wörtern angeben. Dadurch kann für ein Wort beispielsweise bestimmt werden, ob es Teil des Subjekts oder des Objekts eines Satzes ist. Zur Bestimmung von Dependenzen im *TIGER Annotationsschema* [AAB$^+$03] wird Mate Tools [BBHN10] eingesetzt.

Zur Lemmatisierung eines Wortes (z.B. Türme $\rightarrow$ Turm, warte $\rightarrow$ warten) werden Mate Tools und TreeTagger [Sch94] eingesetzt, damit anschließend in dem Tonalitätslexikon SentiWS [RQH10] eine Tonalitätsangabe aus dem Intervall $[-1, 1]$ nachgeschlagen werden kann.

Im folgenden Kapitel werden die untersuchten Aspekte einer Diskussionsverfolgung erläutert. Danach werden zwei Datensätze vorgestellt, die anschließend auf ihren Inhalt und dessen Tonalität untersucht werden. In Kapitel 4 erfolgt ein Vergleich mit verwandten Arbeiten. Abschließend wird ein Fazit gezogen und Ideen für zukünftige Arbeiten aufgeführt.

2 Ansätze zur Diskussionsverfolgung

Bei einer automatisierten Diskussionsverfolgung sind, je nach Analyseziel, ein oder mehrere Aspekte zu berücksichtigen. Im Folgenden werden Überlegungen zur Bestimmung relevanter Sätze angestellt und Ansätze zur Themenextraktion, zum Vergleich semantischer Äquivalenzen und zur Tonalitätsanalyse vorgestellt, deren Ergebnisse anschließend als Argumentationsketten zusammengefasst werden.

2.1 Identifizierung von relevanten Sätzen

In einer Meinungsbildungsanalyse werden Meinungen von Personen in Form von Aussagen untersucht. Daher ist es sinnvoll, nur Sätze zu analysieren, in denen Aussagen gemacht werden. In der deutschen Sprache kann zwischen fünf Satzarten unterschieden werden: Aussagesätzen, Fragesätzen, Ausrufesätzen, Wunschsätzen und Aufforderungssätzen.

Der Inhalt eines Fragesatzes ist für eine Meinungsbildungsanalyse nicht geeignet, da der Autor des Satzes keine Aussage tätigt. Die Zuordnung eines Satzes zu den anderen vier Satzarten ist eine Aufgabe, deren Lösung tiefgehende grammatikalische Eigenschaften benötigt und nicht nur durch das Betrachten des satzbeendenden Zeichens zu lösen ist. Autoren können beispielsweise ein Ausrufezeichen als Satzende eines Aussagesatzes ver-

[1] `https://opennlp.apache.org/`

wenden, in der Absicht dadurch ihrer Aussage mehr Ausdruckskraft zu verleihen. Daher wird vereinfachend angenommen, dass nur mit einem Fragezeichen beendete Sätze ignoriert werden können.

Es gibt Aussagen, für die im gleichen Satz eine Begründung angegeben wird. Diese Begründungen können in einem Nebensatz stehen, der durch eine unterordnende Konjunktion eingeleitet wird. Genauer ausgedrückt durch kausale Konjunktionen (z.B. *weil, da*) bzw. in Kausalsätzen.

Beispiel: $\underbrace{\text{Ich mag dein Auto,}}_{\text{Hauptsatz}}$ $\underbrace{\text{weil es blau ist.}}_{\text{Nebensatz}}$

Eine Möglichkeit, solche Aussagen zu suchen, ist nach Begründungen in einem Nebensatz zu suchen, um anschließend Aussagen aus dem Hauptsatz zu extrahieren. Dazu ist eine Klassifizierung der Satzteile eines gegebenen Satzes mit Kommata in Hauptsätze und Nebensätze hilfreich. Anhand von POS-Tags, Dependenzen und drei Regeln aus Grammatikbüchern (in Bezug auf eine Verberst-, Verbzweit- und Verbletztstellung oder dem Auftreten eines Infinitivs) können Muster identifiziert werden, die im Allgemeinen eine gute Trennung in Hauptsätze und Nebensätze ermöglichen. Anschließend kann nach Begründungen gesucht werden, indem Nebensätze identifiziert werden, die mit einer kausalen Konjunktion beginnen. Häufig vorkommende kausale Konjunktionen sind *da, weil, zumal* und *denn*.

Dieser Ansatz ermöglicht die Identifikation einiger Aussagen, jedoch existieren auch Aussagen in Sätzen, die nicht diesem Schema entsprechen, wie manuelle Stichproben aus den in Abschnitt 3.2 vorgestellten Datensätzen zeigen. Somit ist dieser Ansatz nicht erschöpfend, da er Aussagen, zu denen der Autor keine Begründung angibt, zu denen eine Begründung im nachfolgenden oder vorangestellten Satz erfolgt, Aussagen mit Begründungen innerhalb des Hauptsatzes (z.B. „*Wegen Bauarbeiten ist die Straße blockiert.*“) oder Aussagen mit fehlerhafter Kommasetzung übersieht.

Ohne eine explizite Filterung nach Aussagesätzen muss daher die Annahme getroffen werden, dass in allen Sätzen, die nicht auf ein Fragezeichen enden, Aussagen stecken können und sie folglich potenziell relevant für eine Meinungsbildungsanalyse sind.

2.2 Themenerkennung auf Satzebene

Eine große Aufgabe der maschinellen Sprachverarbeitung ist die Beantwortung der Fragestellung, was das Thema in einem Satz ist bzw. welche Handlung in einem Satz beschrieben wird. Im Folgenden werden zwei Methoden zur Themenextraktion beschrieben.

POS-Tag basiert: Zur Themenerkennung werden alle Wörter eines Satzes extrahiert, deren POS-Tags Nomen (Tag *NN*) oder Eigennamen (Tag *NE*) sind. Dies liefert die Information, wer in einem Satz agiert und womit agiert wird. Dabei werden direkt aufeinanderfolgende Wörter zu einer Fundstelle zusammengefasst, wenn es sich dabei um Eigennamen (Tag *NE*), Nomen aus derselben Nominalphrase (Dependenz *NK*) oder um Namensbestandteile (Dependenz *PNC*) handelt. Ein schlichtes Zusammenfassen von aufeinander-

folgenden Wörtern kann zu Problemen führen, wie der Satz „*Peter mag Kinder, die mit Feuerwehrwagen Rettungen nachspielen.*“ zeigt, da dort die aufeinanderfolgenden Nomen *Feuerwehrwagen* und *Rettungen* fälschlicherweise zusammengefasst werden.

Dependenzbasiert: Da bei dem POS-Tag basierten Ansatz nicht die Information berücksichtigt wird, welche Aktion im Satz beschrieben wird, bezieht der zweite Ansatz zur Themenextraktion die Rolle des Verbs mit ein. In der grundlegendsten Version wird dabei nach SVO-Tripeln, die jeweils aus drei Wortgruppen bestehen und damit zweistellige Valenzen darstellen, gesucht. Die erste Wortgruppe repräsentiert das Subjekt im Satz, die zweite das Verb und die dritte das Objekt. Identifiziert wird ein SVO-Tripel durch ein Verb, von dem eine Dependenz für das Subjekt $s \in \{\text{SB,SBP,SP}\}$ und eine Dependenz für das Objekt $o \in \{\text{OA,OA2,OG,OP,DA}\}$ ausgehen. Aus dem Beispielsatz „*Peter kritisiert Anne.*“, dessen Dependenzen in Abbildung 1 visualisiert sind, wird das SVO-Tripel *(Peter, kritisiert, Anne)* extrahiert.

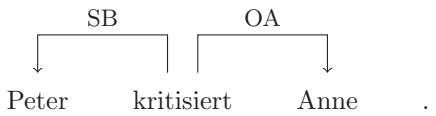


Abbildung 1: Beispiel für die Dependenzen eines SVO-Tripels

Bei SVO-Tripeln sollte darauf geachtet werden, dass Subjekte und Objekte grammatikalische Funktionen sind und nicht semantische Rollen darstellen. In Aktivsätzen gilt in der Regel, dass das Subjekt mit dem Agens (der handelnden Entität) und das Objekt mit dem Patiens (der betroffenen Entität) übereinstimmen. In Passivsätzen kann eine Rollenvertauschung auftreten, wie z.B. im Satz „*Markus wird von Peter bestohlen.*“, in dem *Markus* als Subjekt und als Patiens agiert und *Peter* das Objekt und den Agens verkörpert. Ebenfalls müssen Negationen gesondert behandelt werden. Verben mit einer anderen Valenzordnung werden Gegenstand zukünftiger Forschung sein.

2.3 Semantische Äquivalenzen erkennen

Nachdem ermittelt wurde, worüber in einzelnen Sätzen gesprochen wird, erfolgt anschließend eine Überprüfung, ob sich verschiedene Wörter auf den gleichen Sachverhalt beziehen bzw. ob semantische Äquivalenzen bestehen, da sprachlich vielfältige Möglichkeiten bestehen, denselben Inhalt auszudrücken. Wurde als Thema eines Satzes ein Nomen extrahiert, so gilt zu identifizieren, ob das Thema synonym zu einem anderen Thema ist, um Aussagen über dasselbe Thema gruppieren zu können. Dazu kann ein Thesaurus verwendet werden, in dem üblicherweise sogenannte Synsets enthalten sind. Ein Synset kann aus mehreren Wörtern bestehen, welche die gleiche semantische Bedeutung tragen wie beispielsweise $\{$*Orange, Apfelsine*$\}$.

Als deutschsprachiger Thesaurus wird GermaNet [HF97] verwendet, in dem Synsets für Nomen, Verben und Adjektive vorhanden sind. Für Nomen sind in GermaNet 9.0 insgesamt 71575 Synsets vorhanden, mit denen überprüft werden kann, ob zwei Nomen syn-

onym zueinander sind.

Es ist geplant, weiterführende Techniken der *Koreferenzanalyse* (Zusammenfassung verschiedener Bezeichnungen für dieselbe Person bzw. dasselbe Objekt) und der *word sense disambiguation* (Erkennung der kontextabhängigen Wortbedeutung) in nachfolgenden Arbeiten zu berücksichtigen.

2.4 Tonalitätsbestimmung

Für die extrahierten Themen wird anschließend untersucht, mit welchen Wörtern der jeweilige Sachverhalt beschrieben bzw. welche Tonalität dabei ausgedrückt wird. Dazu werden von den gefundenen Nomen und den voranstehenden Adjektiven die wortbasierten Tonalitätsangaben in SentiWS nachgeschlagen und zu einem Mittelwert zusammengefasst. Bei SVO-Tripeln wird zusätzlich das Verb einbezogen. Adjektive werden unter anderem für die Tonalitätsbestimmung bei Produktrezensionen [HL04] berücksichtigt. Im Rahmen dieser Arbeit wird untersucht, inwiefern Adjektive zur Tonalitätsbestimmung von Diskussionsbeiträgen beitragen.

Sprachliche Negationen können auf verschiedene Arten ausgedrückt werden. Negationen durch die Verwendung eines Affixes (z.B. *glücklich* und *unglücklich*) werden teilweise durch SentiWS abgedeckt, wenn für beide Wörter Einträge vorhanden sind. Bei einer Negation durch das Wort *„nicht“* wird vereinfachend angenommen, dass die Polarität eines Adjektivs umgedreht wird, wenn das Wort *„nicht“* direkt vorangestellt ist. Alternativ kann überprüft werden, ob für ein negiertes Adjektiv in GermaNet per Antonym-Beziehung ein entgegengesetztes Adjektiv existiert, das in SentiWS eine Tonalitätsangabe besitzt.

2.5 Argumentationsketten

Nachdem bereits Themen identifiziert und dazu Tonalitätsangaben ermittelt wurden, erfolgt eine Gruppierung in Argumentationsketten, die sich jeweils auf genau ein Thema, wie eine Person oder einen Begriff, beziehen. Eine Argumentationskette kann als ein Zeitstrahl betrachtet werden, auf dem die zu einem Thema gemachten Äußerungen mit den zugehörigen Tonalitäten zeitlich geordnet sind.

Die Konstruktion von Argumentationsketten aus Sätzen erfolgt in mehreren Schritten (vgl. Abbildung 2):

(1) Für den durch die NLP-Pipeline aufbereiteten Eingabesatz erfolgt eine Themenerkennung.

(2) Für alle gefundenen Themen wird eine Tonalität bestimmt.

(3) Für jede Fundstelle wird überprüft, ob eine Argumentationskette existiert, deren Thema textuell mit der Fundstelle übereinstimmt oder dazu per GermaNet synonym ist. Kann keine übereinstimmende Argumentationskette gefunden werden, wird eine

neue Argumentationskette begonnen.

(4) Der in Schritt (3) bestimmten Argumentationskette wird ein neues Element angefügt, in dem das Thema, die dazugehörige Tonalität und Metadaten (Erstellungsdatum, Autorname, Referenz auf den Satz) vermerkt werden.

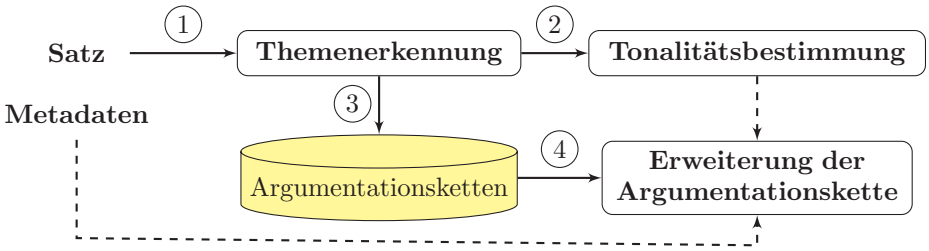


Abbildung 2: Konstruktion von Argumentationsketten

3 Auswertung

Die genannten Ansätze zur Diskussionsverfolgung werden im Rahmen einer prototypischen Implementierung auf zwei Datensätze angewendet. Dabei werden exemplarisch mehrere Analysekriterien definiert.

3.1 Analysekriterien

Abhängig von der Aufgabenstellung einer Meinungsbildungsanalyse können unterschiedliche Betrachtungen stattfinden. Eine Betrachtung kann globale Eigenschaften der überwachten Texte in Hinblick auf eine Kommentarfrequenz und auf durchschnittliche Tonalitätswerte untersuchen oder auf themenspezifischen Eigenschaften der konstruierten Argumentationsketten erfolgen. Bei einer längerfristigen Überwachung eines Themas ist eine zeitliche Gruppierung von Texten sinnvoll. Eine abschnittweise Untersuchung ermöglicht einen Vergleich zwischen Zeitabschnitten und dadurch die Analyse einer zeitlichen Entwicklung. Die zeitliche Gruppierung kann z.B. per fester Zeitspanne, per fester Anzahl an Kommentaren oder einer Kombination beider Kriterien erfolgen.

Bei einer Meinungsbildungsanalyse ist ebenfalls interessant zu bestimmen, über welche Themen in den analysierten Texten gesprochen wird. Anhand der Länge aller Argumentationsketten können häufig besprochene Themen identifiziert werden. Bei der Betrachtung eines konkreten Themas kann untersucht werden, wann es häufig besprochen wird und wie sich die verwendete Tonalität entwickelt.

Firmen können beispielsweise aus der automatisierten Analyse von Social Media einen Vorteil ziehen, wenn sie maschinell Meinungen über ihre eigenen Produkte untersuchen.

Wird über ein bestimmtes Produkt innerhalb kürzester Zeit vermehrt Kritik geäußert, so können die Entwicklungs- und Marketingabteilungen auf ein mögliches Problem des Produkts hingewiesen werden.

3.2 Datensätze

Die Ansätze zur Diskussionsverfolgung werden auf zwei Datensätze angewendet. Der erste Datensatz wird als *HHU Normsetzungskorpus* bezeichnet. Er besteht aus 434 Kommentaren (1444 Sätzen), die im Rahmen der *Online-Fakultätsratssitzung „Neugestaltung der Promotionsordnung“*[2] [EST+14] auf einer Instanz der Plattform *Adhocracy*[3] abgegeben wurden. Der zweite Datensatz wurde durch einen Crawling-Prozess aus Kommentaren eines deutschsprachigen Nachrichtenportals zusammengestellt und wird als *Krim-Korpus* bezeichnet. Er besteht aus 62362 Kommentaren (354476 Sätzen), die aus Nachrichten der Kategorie *Politik* stammen, die mit dem Schlagwort *Krim* versehen sind und im Zeitraum vom 1.2.2014 bis zum 31.7.2014 abgegeben wurden.

3.3 Analyseergebnisse

Die konstruierten Argumentationsketten wurden auf Gesprächsthemen und auftretende Tonalitäten untersucht. Dazu wurden die Kommentare des Krim-Korpus wochenweise gruppiert. In der Woche des 17.3.2014, in der ein Referendum über die Zugehörigkeit der Halbinsel Krim zu Russland oder zur Ukraine erfolgte, wurden mit über 9000 Kommentaren die meisten Kommentare in einer Woche gemacht. Für jede untersuchte Woche ist der durchschnittliche Tonalitätswert leicht negativ.

Gesprächsthemen: In Tabelle 1 werden die zehn häufigsten Themen, die über den POS-Tag basierten Ansatz ermittelt wurden, mit der Länge der jeweiligen Argumentationskette angegeben.

Platzierung	HHU Normsetzungskorpus	Krim-Korpus
1	Überprüfung (106)	Russland (23436)
2	Arbeit (105)	Ukraine (18758)
3	Promotion (95)	Putin (13720)
4	Publikation (88)	Krim (11374)
5	Betreuer (67)	USA (11029)
6	Gebiet (67)	Grund (10811)
7	Vorschlag (62)	Westen (8995)
8	Beispiel (51)	EU (8487)
9	Note (51)	Mensch (6405)
10	Gutachter (47)	Regierung (6271)

Tabelle 1: Liste der zehn häufigsten Themen der beiden Datensätze

[2] `https://normsetzung.cs.uni-duesseldorf.de/`
[3] `https://adhocracy.de/`

Eine zu Beginn gemachte, intuitive Annahme, dass reale Sachverhalte von Ereignissen aus der Region Ukraine im Krim-Korpus häufig durch das gleiche SVO-Tripel ausgedrückt werden, konnte nicht bestätigt werden. Anstatt dass in häufig genannten Tripeln die Beziehung zwischen zwei Personen, wie z.B. *(Peter, kritisiert, Anne)*, ausgedrückt wird, besitzen die am häufigsten auftretenden Tripel das Wort „*Ich*“ als Subjekt oder Verben der Meinungsäußerung wie „*denken*“ oder „*meinen*“.

Tonalität: Zunächst ist anzumerken, dass für beide Korpora durch das Tonalitätslexikon SentiWS nur wenige Tonalitätsangaben pro Satz bestimmt werden konnten. Im Krim-Korpus gibt es lediglich 0,9 Tonalitätsangaben pro Satz. Beim HHU Normsetzungskorpus sind es 1,2 annotierte Tokens pro Satz. Dies erschwert die Tonalitätsbestimmung einzelner Themen, da durch die niedrige Anzahl an Angaben oft eine neutrale Tonalität angenommen werden muss. Wird als Schwellwert $\alpha = 0{,}1$ zur Einordnung eines numerischen Tonalitätswerts in $t \in \{\text{positiv, neutral, negativ}\}$ verwendet, so tritt für das im Krim-Korpus am häufigsten auftretende Thema *Russland*, unter Berücksichtigung von vorangestellten Adjektiven, in 99,5% eine neutrale Tonalität auf.

Da vergleichbare Verteilungen auch bei anderen Themen entstehen, ist anzunehmen, dass ein größeres Tonalitätslexikon benötigt wird und dass Tonalitäten im Krim-Korpus auch durch andere sprachliche Mittel als Adjektive ausgedrückt werden. Daher werden sich zukünftige Arbeiten mit der Frage beschäftigen, welche sprachlichen Mittel im Rahmen von Online-Diskussionen besonders viel Tonalität tragen und wie diese automatisiert erfasst werden können.

Zur approximativen Untersuchung einer zeitlichen Tonalitätsentwicklung wird daher auf die Annahme zurückgegriffen, dass die Tonalität einer Fundstelle durch eine Mittelwertbildung aller Tonalitätsangaben des gesamten Satzes angegeben wird. Die zeitliche Entwicklung der Tonalitäten wird nur anhand des Krim-Korpus untersucht, da der HHU Normsetzungskorpus zu klein ist. Für das Thema *Russland* verschiebt sich die Verteilung der Tonalität zu 9,7% positiv, 67,7% neutral und 22,6% negativ. In Abbildung 3 werden exemplarisch die Tonalitäten der Argumentationskette über das Thema *Wahl* des Krim-Korpus (jeweils als Mittelwert einer wochenweisen Gruppierung) dargestellt. Die dort auftretenden Vorzeichenwechsel zwischen zwei Zeitpunkten können im Rahmen einer automatisierten Themenüberwachung interessant sein.

4 Verwandte Arbeiten

In [KG12] wird ein verborientierter Ansatz zur Tonalitätsbestimmung von Meinungen in englischsprachigen Diskussionen vorgestellt. Die Autoren erstellen ein Tonalitätswörterbuch aus 440 Verben, die eine Meinung ausdrücken können, wie z.B. *agree, hate, like* und *love*, zu denen jeweils die Größe des Valenzrahmens und eine numerische Tonalität angegeben ist. Anhand eines Verbs wird entschieden, wie sich die Tonalitäten der Wörter im Valenzrahmen aufeinander auswirken. Die Autoren bemerken, dass in den untersuchten Kommentaren 47% der Tonalität durch Verben und nur 33% der Tonalität durch Adjektive ausgedrückt werden. Sie kommen ebenfalls zu dem Fazit, dass die Übertragung von Tech-

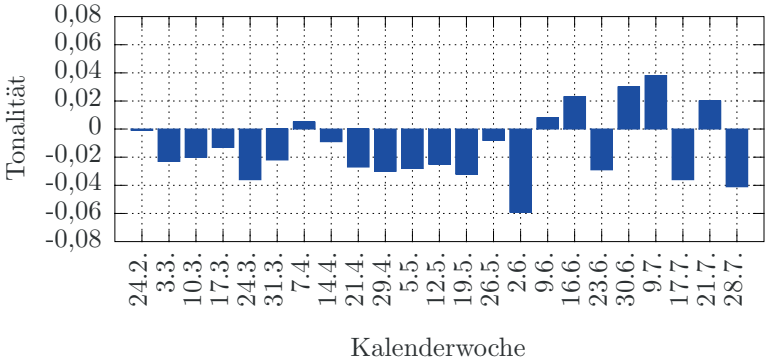


Abbildung 3: Tonalitätsentwicklung der Argumentationskette über das Thema *Wahl*

niken zur Tonalitätsbestimmung von Produktrezensionen auf Online-Diskussionen problematisch ist.

Eine POS-Tag basierte Themenerkennung, die Nomen als Themen identifiziert, wird in [CZKL10] verwendet. Die Autoren extrahieren politische Haltungen aus Aussagen von amerikanischen Senatoren und benutzen für die Tonalitätsbestimmung Adjektive, Verben und Adverbien.

5 Fazit und Ausblick

Im Rahmen dieser Arbeit wurden verschiedene Aspekte einer maschinellen Analyse von deutschsprachigen Online-Diskussionen vorgestellt. Dabei wird das Fazit gezogen, dass ein valenzorientierter Ansatz, der sich auf Verben als zentrales Element eines Satzes bezieht, detaillierter untersucht werden sollte. Dazu wird ein elektronisches Valenzwörterbuch, in dem pro Verb die Größe des Valenzrahmens angegeben ist, erforderlich sein. Zur Tonalitätsanalyse wird in zukünftigen Arbeiten ein individuell auf die Analyse von Online-Diskussionen angepasstes Verfahren entwickelt werden, das vor allem die Rolle von Verben der Meinungsäußerung stärker einbezieht und dazu auf ein noch zu entwickelndes Tonalitätslexikon zurückgreifen wird.

Darüber hinaus werden Textbeiträge aus einem Online-Partizipationsverfahren annotiert werden, auf denen die vorgestellten Aspekte genauer evaluiert werden können. In Zukunft wird geprüft werden, ob OpenThesaurus[4] als weiterer deutschsprachiger Thesaurus und Wiktionary[5] als zusätzliches Wörterbuch zur Grundformreduktion eingesetzt werden können. Darüber hinaus wird die Zuordnung von Aussagen zu einer konkreten Argumentationskette durch Techniken der Koreferenzanalyse, der Anaphorikauflösung und der word sense disambiguation verbessert werden.

[4] https://www.openthesaurus.de/
[5] https://de.wiktionary.org/

Literatur

[AAB+03] S Albert, J Anderssen, R Bader, S Becker, T Bracht, S Brants, T Brants, V Demberg, S Dipper, P Eisenberg, S Hansen, H Hirschmann, J Janitzek, C Kirstein, R Langner, L Michelbacher, O Plaehn, C Preis, M Pussel, M Rower, B Schrader, A Schwartz, Smith G und H Uszkoreit. TIGER-Annotationsschema. Bericht, Universität Potsdam, Universität Saarbrücken, Universität Stuttgart, 2003.

[BBHN10] Anders Björkelund, Bernd Bohnet, Love Hafdell und Pierre Nugues. A High-Performance Syntactic and Semantic Dependency Parser. In *COLING (Demos)*, Seiten 33–36. Demonstrations Volume, 2010.

[CZKL10] Bi Chen, Leilei Zhu, Daniel Kifer und Dongwon Lee. What Is an Opinion About? Exploring Political Standpoints Using Opinion Scoring Model. In *AAAI*. AAAI Press, 2010.

[EST+14] Tobias Escher, Jost Sieweke, Ulf Tranow, Simon Dischner, Dennis Friess, Philipp Hagemeister und Katharina Esau. Internet-Mediated Cooperative Norm Setting in the University: Design and Evaluation of an Online Participation Process to Redraft Examination Regulations. In *IPP2014: Crowdsourcing for Politics and Policy*, 2014.

[HF97] Birgit Hamp und Helmut Feldweg. GermaNet - a Lexical-Semantic Net for German. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Seiten 9–15, 1997.

[HL04] Minqing Hu und Bing Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, Seiten 168–177. ACM, 2004.

[KG12] Mostafa Karamibekr und Ali A. Ghorbani. Sentiment Analysis of Social Issues. In *Proceedings of the 2012 International Conference on Social Informatics*, SOCIALINFORMATICS '12, Seiten 215–221. IEEE Computer Society, 2012.

[RQH10] R. Remus, U. Quasthoff und G. Heyer. SentiWS – a Publicly Available German-language Resource for Sentiment Analysis. In *Proceedings of the 7th International Language Resources and Evaluation (LREC'10)*, Seiten 1168–1171, 2010.

[Sch94] Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, 1994.

[STST99] Anne Schiller, Simone Teufel, Christine Stöckert und Christine Thielen. Guidelines für das Tagging deutscher Textcorpora mit STTS (kleines und großes Tagset). Bericht, Universität Stuttgart, Universität Tübingen, Stuttgart, Germany, 1999.

Development and Evaluation of a Facebook-based Product Advisor for Online Dating Sites

Martin Winter[1], Sebastian Goebl[1], Nina Hubig[2], Christopher Pleines[3] and Christian Böhm[1]

[1]University of Munich, Germany,
winterm@cip.ifi.lmu.de, {goebl,boehm}@dbs.ifi.lmu.de
[2]Helmholtz Zentrum Munich, Germany,
nina.hubig@helmholtz-muenchen.de
[3]Zu-Zweit.de, pleines@zu-zweit.de

Abstract: Social networks play an important role in Web 2.0. For many users establishing contacts and staying in touch in the virtual world is more than just a spare time filler. In social networks like Facebook they provide much information about themselves in user profiles. Also for online dating the focus is on establishing new contacts. In general, three types of dating sites can be distinguished: more serious dating agencies, less focused singles' platforms, and casual dating sites. In the proposed paper we develop a product advisor that uses the Facebook profile information provided by a user to classify her or him to one of the three dating site categories which is most suitable for the user's purpose. For classification we use Naive Bayes. To train the classifier we investigate the correlation between profile information and choice of dating sites. We also evaluate this correlation on collected data of representative German dating sites. Although a sharp distinction is hard to find, tendencies and enlightening insights are revealed by the collected data.

1 Introduction

Since the evolution from Web 1.0 to 2.0 the focus is no longer exclusively on providing information but on involving people. Social networks represent a central building block for Web 2.0, providing virtual communities for social relations online. In Germany, social networks are an essential part of life. With 46.1 million of 81.8 million inhabitants more than half of Germany uses social networks, where Facebook is leading with 13 million users per day (24 million users in total) far ahead of other social networks like `Ask.fm` (0.6 million users per day) or `Xing.com` (0.4 million users per day) [MB13]. Sharing news with friends is a quick and easy thing to do in social networks. This simplicity helps to keep information up to date. An immense data pool has been formed by all this profile information. It is used e.g. for target group oriented advertising. Also, developers of applications for Facebook can access the complete profile information if a user gives her or his consent. This is what we make use of in the development of our product advisor for online dating sites.

We distinguish three categories of online dating sites. The first category are *dating agen-*

Dating Agencies	Singles' Platforms	Casual Dating Sites
Parship.de	Neu.de	Secret.de
Elitepartner.de	DatingCafe.de	C-Date.de
eDarling.de	FriendScout24.de	Lovepoint.de

Table 1: Dating sites selected for our product advisor.

cies. These are aimed at users who are interested in serious relationships. The sites make use of psychological questionnaires to find a matching partner. Then, there are less focused *singles' platforms* for users who want to flirt and date as well — but with not as serious intentions as dating agencies' users. Finally, *casual dating* sites aim at users who are looking for erotic adventures without being interested in a relationship. For the proposed product advisor for online dating sites we select three German dating sites for each category, shown in Table 1.

User questionnaires on dating agencies' websites aim to collect information about the user to offer him the optimal matching partner. We go a step further and analyze user interests already before she or he even chooses a specific dating website and can be confronted with a questionnaire. Also, by using Facebook user profile information we access an even larger data pool as is possible by questionnaires. The dating agency Parship.de ask every user 70 questions for matching purposes. On the other hand, in average our Facebook-based product advisor collects 124 likes per user, which is used for classification.

Our product advisor is trained by user decisions: in the data collecting phase it has suggested three random dating sites — one of each of the dating site categories. The user's decision to access a dating site (by clicking on one of the three suggested links) as well as information about him signing up has been stored together with her or his Facebook profile information. Using Naive Bayes we construct a classifier for each dating site category. The classifier decides which of the three dating site categories is most suitable for the user. We build the classifier based on the preferences of dating site categories that have been gained in the data collecting phase. Using cross validation the collected data is also used to test the quality of our classifier. Additionally, we present insights into interesting correlations that have been discovered when analyzing the data.

2 Related Work

Research has addressed social networks in general and Facebook especially, mostly concerning privacy issues as well as users' generosity of giving away private information. Stutzman et al. [SGA12] investigated from 2005 to 2011 how Facebook users handle their own privacy. It was found out that exposure of private information to the public has decreased during this period, while on the other hand sharing information to Facebook friends has increased. Privacy issues and the amount of shared information were investigated by Gross et al. [GA05] on 4000 students of the Carnegie Mellon University. Especially, attacks on user like stalking were considered. It was found out that users pro-

vide information to non-direct 'friends' which in fact are total strangers. Facebook's Gross National Happiness Index investigated the connection between positive and negative status updates, and the mood of the population of different countries. It was analyzed by Wang et al. [WKSR14] who found out that it does not sufficiently express the nations' mood. Kosinski et al. [KSG13] drew attention to their study on predicting private attributes from Facebook information. By only few information about Facebook likes it was possible to construct an exact personal profile about Facebook users. In the study 88% of the male users were classified into their correct sexual orientation and 82% of them properly distinguished between Christians and Moslems. The authors developed the Facebook application 'You are what you like' allowing users to receive a personality analysis based on their Facebook likes. In average Facebook users provided more than 200 likes on their Facebook profile. Bachrach et al. [BKG+12] also addressed the possibility to draw conclusions from Facebook likes to the user's personality. For an exact personality analysis the 'Big Five' personality traits (openness, conscientiousness, extroversion, agreeableness, and neuroticism) were reconstructed for each user. As result it was proved feasible to extract the Big Five from the provided personal information. By asking the user to fill out a questionnaire online dating agencies follow the same intention to analyze the user's personality in order to provide optimal matches. However, these algorithms are not made public. The related work concentrates on extracting personality traits from user information provided more or less intentionally through Facebook profile information. The mentioned related work proves that drawing conclusions from Facebook profile information is possible. We tie on this research and use the user-provided Facebook information to implicitly also analyze his personality. We aim to develop a product advisor that suggests the online dating site category that is most suited to his character as far as it is reconstructed from the provided information. As an additional benefit the training of the product advisor allows an evaluation of which types of personality structure are interested in which type of only dating site category. This represents an insight which can be used to tailor online dating sites to their target group.

3 The Dating Barometer

Facebook offers the possibility to create web applications providing tools for Facebook developers. These applications (called *apps*) are available for Facebook users only. When accessing an app for the first time, the user is asked if he agrees in sharing his profile information with the Facebook application. In case of consent the app has access to private profile information. Our application is called *Dating Barometer* (as it measures dating intentions) and is designed for German users, since it is comparing German online dating sites. First, training data for the product advisor needs to be collected to enable classification. Therefore, the Dating Barometer randomly creates a suggestion of dating sites for the user. The suggestion consists of a ranking of the three dating site categories represented by one of the sites as assigned to in Table 1 (cf. Figure 1). Although ordered randomly for collecting training data, the final product recommendation based on Facebook profile information is presented in the same way. The data collected by the Dating

Figure 1: Dating Barometer creating a ranked suggestion of dating sites for the user.

Barometer consists basically of two parts: the *general Facebook profile information* and the *extended Facebook profile information*. The first comprises *name*, *e-mail address*, *sex*, *birthday*, *place of residence*, and *profile picture*. The second comprises additional profile information about *activities*, *interests*, *favorite music*, *favorite TV shows* and *favorite movies*, as well as contents that are connected by pressing the like button, called Facebook *likes*. Additional profile information about *age*, *interest in men and/or women*, and *relationship status* is expected to be very relevant and, thus, is collected, too. We name the extended Facebook profile information *meta (profile) information* in contrast to *basic* rest of the profile information.

The Dating Barometer is implemented in PHP using the PHP-SDK provided by Facebook. All data is stored in a MySQL database. On his first visit a user receives a HTTP cookie containing a distinct identifier. This is necessary, since the Dating Barometer has no profile information at all, before the user consents to sharing. Thus, if a user visits the Barometer but leaves without sharing and revisits again, only a cookie can identify him. All user interaction with the Dating Barometer is stored in our database. New users or revisiting users who have not shared profile information are welcomed by our virtual character Lisa and informed that the application offers a recommendation for online dating sites based on his profile information. If consenting the user is given a random recommendation as defined above. Then, the user can ask for further information about a specific dating site and

receives a test review (button “Testbericht lesen”(read review) on Figure 1), or he can directly access the registration form for the dating site (button “Kostenlos ausprobieren”(try for free) on Figure 1). Clicking one of these buttons is recorded for this user in our database as a *click-out*. In case the user signs up for the online dating site, this is stored as a *sign-up* for this user.

4 Evaluation

For about three months the Dating Barometer has been advertised to attract users and been active for collecting data on Facebook. Additional to the user information on Facebook, the accessed online dating sites (cf. Table 1) reported if a click-out resulted in a sign-up. In this section we present general observations derived from the collected data and construct a Naive Bayes classifier that predicts a dating site category for a user providing her or his Facebook information. We used LIBSVM for the Naive Bayes classifier.

4.1 General Observations

4.1.1 Conversion Rate

Until the end of the e n guided to the Dating Barometer. As Figur onsent when asked for

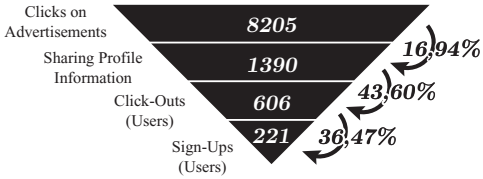


Figure 2: Conversion rate of users visiting the Dating Barometer.

profil information. With 896 click-outs by 606 different users the rate is 43.60% with a mean of 1.48 click-outs per user. This means that nearly the half of the users sharing profile information also made used of the Dating Barometer. 221 (36.47%) of these users also signed up at an online dating site.

4.1.2 Gender

The Dating Barometer attracted men more then women. Only 15.97% of all users sharing profile information were female users (cf. Figure 2). Of these female users 31.08% resulted in a click-out and 10.36% in a sign-up. For men the relative click-out and sign-up rates are higher: 45.97% clicked on a dating site suggesting and 16.98% signed up for a dating site.

	Total	Female	Male	Unknown
Clicks on Advertisements	8205	—	—	—
Sharing Profile Information	1390	222	1166	2
Click-Outs	606	69	536	1
Sign-Ups	221	23	198	0

Table 2: Gender of Dating Barometer users.

4.1.3 Age

The mean age of all Dating Barometer users was 31.12 ± 11.35 years. Most of the users (52.37%) were between 18 and 29 years of age (cf. Figure 3). 23.60% of the users were

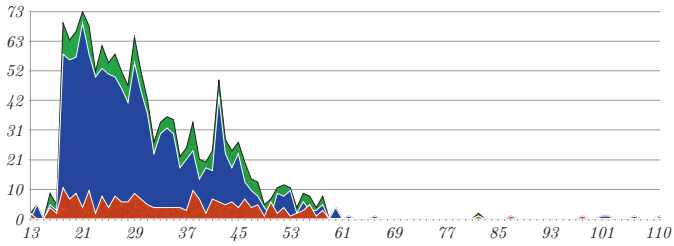


Figure 3: Total number of users (vertical axis) and their age (horizontal axis) for female (red), male (blue) and all (green) users.

between 30 and 39 years, whereas 16.12% of the users were between 40 and 49 years. The remaining users were over 50 years, constantly decreasing. Some high ages might not be too realistic. However, they do not appear to impact the overall picture. The male users' age distribution corresponds to the overall distribution, whilst the distribution of female users between 18 and 55 is relatively uniform and only slightly decreasing. Users under 18 years of age are not regarded in the following observations as they are not allowed to use most of the online dating sites because of the sites' general terms and conditions.

Meta Profile Information	Mean	Minimum	Maximum
Likes	124.03 ± 306.69	0	4552
TV Shows	0.03 ± 0.17	0	1
Activities	2.97 ± 8.78	0	213
Interests	1.48 ± 2.87	0	36
Music	0.50 ± 0.95	0	8
Movies	0.03 ± 0.17	0	1
Total	129.04 ± 289.49	0	4621

Table 3: Mean, minimal and maximal number of provided meta profile information entries for all Dating Barometer users (who consented to sharing profile information).

4.1.4 Meta Profile Information

For every meta profile information there are users with no details. 32.59% users have provided no information to Facebook for at least one meta information. A total of 61 users have given no meta information at all. Table 3 shows mean, minimal and maximal number of entries of meta information for all users who share information.

4.1.5 Relationship Status

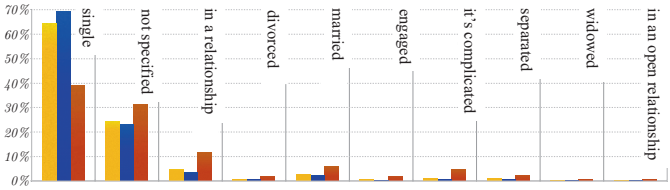
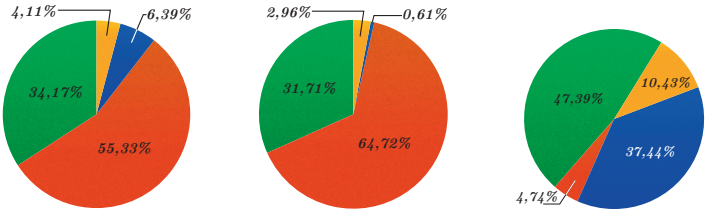


Figure 4: Occurrences of types of relationship status under Dating Barometer users showing male (blue), female (red) and all (yellow) users.

63.09% of all Dating Barometer users declared themselves as being single, followed by 23.81% not specifing their relationship status. All other types of relationship status were represented by maximal 4.53% of the Dating Barometer users (cf. Figure 4).

4.1.6 Sexual Orientation



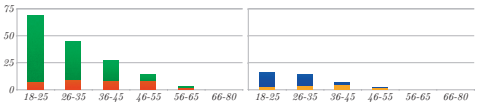
(a) Total (b) Male Users (c) Female Users.

Figure 5: Sexual Orientation: interest in men (blue), women (red), both (yellow) or not specified (green).

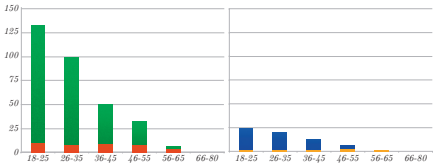
Illustrated by Figure 5, most of the Dating Barometer users state being heterosexual. Women tend to declare this less (37.44%) than men (64.72%). A total of 4.11% states to be bisexual, women a bit more often than men. Interest in the same sex is told by 4.74% of the female and 0.61% of the male users. 34.17% of all users have not specified their sexual orientation.

4.1.7 Distribution of Age Groups over Online Dating Site Categories

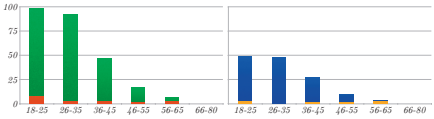
Figure 6 distinguishes the age groups related to the three online dating site categories. The charts also differentiate between click-outs and sign-ups for male and female users. Some results can already be read: singles' platforms were most successful in click-outs (23.53%), while casual dating sites showed the highest sign-up rate (9.86%). Dating agencies were in click-outs (11.73%) and sign-ups (2.88%) at the end. 4.60% of the users signed up at a singles' platform, while 18.92% clicked in casual dating sites. Singles' platforms and dating agencies both have a strong peak in the 18-25 age group. Instead, casual dating sites are nearly as popular in the 26-35 age group.



(a) Dating Agencies



(b) Singles' Platforms



(c) Casual Dating Sites

Figure 6: Distribution of click-outs/sign-ups (total number: vertical axis) over age groups (horizontal axis) for male click-outs (green), male sign-ups (blue), female click-outs (red) and female sign-ups (yellow) for the three different online dating site categories.

4.2 Naive Bayes Classification

4.2.1 Data Selection

To identify relevant meta profile information and to exclude unspecific data, e.g. likes as 'McDonalds' or 'CocaCola', we calculate the relevance of meta information. We derive it from the relation of Dating Barometer users *liking* a meta profile information (MPI) to all Facebook users *liking* it as

$$relevance_{MPI} = \frac{\textit{\# of Dating Barometer users with MPI}}{\textit{\# of all Facebook users with MPI}}.$$

For data classification we only use the top 20 most relevant meta profile information entries.

We address another not trivial problem: meta data is generated by and, thus, named by the Facebook users. Therefore, for the same subject there exist several different meta information entries. This can be due to spelling mistakes or to different levels of concept

hierarchies, e.g. 'The Simpsons' compared to 'Homer Simpson' or 'Bart Simpson', all relating to the same TV series. However, other meta information entries are not connected in any way even if written in a similar way. E.g., 'Jessica Simpson' is equal to the TV series if comparing text strings. To deal with this problem we use the rich possibilities of Google's search. Google's syntax recognition allows to differentiate between search keywords by using the top 100 URLs. On this basis, we construct a Google crawler that fills a database with the top 100 URLs for each meta profile information entry.

After crawling the search keywords, a main URL is set for each of the top 20 meta information entries. By comparing strings of meta information entries in combination with comparing the top 100 URLs of each entry we associate the top 20 meta information entries with its matching entries and use the combined new meta information entries as input data for our evaluation. To tie on the example from above, a Google search for the TV series meta information entry 'The Simpsons' retrieves the URL `www.thesimpsons.com` as first search result. This URL is also suitable to be set as main URL for this entry. If searching for 'Bart Simpson', the same URL appears as tenth search result. On the other hand, a search for 'Jessica Simpson' does not show this URL at all in the top 100 results. Therefore, defining similarity over Google search results is a promising way to group meta information.

Details to religion are free text fields where users can describe their attitude using own words. In order to make use of this information, the main religious orientations in Germany, *Catholicism*, *Protestantism*, *Judaism* and *Islamism*, were extracted using string comparison. Altogether, for each user we extract the attributes *gender*, *age*, *sexual orientation*, *relationship status*, *religion*, *spoken language(s)*, *operating system*, *education*, *top 20 meta likes* and the connected *click-outs* and *sign-ups*.

4.2.2 Naive Bayes Classifier

We create a Naive Bayes classifier for a simplified data set using all but meta profile information. The numbers of click-outs (606) and sign-ups (221) suggest using k-fold cross validation with $k = 10$ to validate our classifier. Table 4 presents the confusion matrix for the Naive Bayes classifier for click-outs trained on the simplified data set. For

DA	**SP**	**CD**	
28	129	6	**DA**
35	234	11	**SP**
11	139	13	**CD**

Table 4: Confusion matrix for click-out classifier for all dating site categories: dating agencies (DA), singles' platforms (SP) and casual dating sites (CD)

dating agencies and casual dating sites the negative predictive value is 74.62% and 73.76%. The predictive value for singles' platforms is 55.77%. The weighted mean of the negative predictive value of this classifier is 65.73%.

We also construct a Naive Bayes classifer constructed for sign-ups using all profile information including meta information. It is validated to a negative predictive value showing

a weighted mean of 68,54%.

5 Conclusion

The click-outs showed that the Dating Barometer users in the age groups 18-25 and 26-35 are more interested in singles' platforms and casual dating sites than dating agencies. However, singles' platforms and dating agencies show about the same number of sign-ups in these age groups, but the number of sign-ups in casual dating sites is more than double the number. Thus, this means that these users are more interested in erotic adventures.

With altogether 1390 users of which could be used only 44% for click-out classification and 16% for sign-up classification, the amount of used data is rather small. But still, results clearly show facts as such that dating agencies are also most relevant for the age group 18-25. We can also define with an accuracy of 81% if users sign up in a dating agency or not. The accuracy for singles' platforms here is 71% and for casual dating sites 67%.

However, classification by Naive Bayes classifiers to predict the type of dating site proves to be a difficult task. However, it is also due to the small numbers of click-outs and sign-ups that it is hard to establish useful classifiers.

References

[BKG+12] Yoram Bachrach, Michal Kosinski, Thore Graepel, Pushmeet Kohli, and David Stillwell. Personality and Patterns of Facebook Usage. In *Proceedings of the 3rd Annual ACM Web Science Conference*, WebSci '12, pages 24–32, New York, NY, USA, 2012. ACM.

[GA05] Ralph Gross and Alessandro Acquisti. Information Revelation and Privacy in Online Social Networks. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, WPES '05, pages 71–80, New York, NY, USA, 2005. ACM.

[KSG13] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 2013.

[MB13] Ayaan Mohamud and Berit Block. 2013 Future in Focus - Digitales Deutschland. `http://www.comscore.com/ger/Insights/Presentations_and_Whitepapers/2013/2013_Future_in_Focus_Digitales_Deutschland`, 2013. Accessed: 2014-03-26.

[SGA12] Fred Stutzman, Ralph Gross, and Alessandro Acquisti. Silent listeners: The evolution of privacy and disclosure on Facebook. *Journal of privacy and confidentiality*, 4(2):7–41, 2012.

[WKSR14] N. Wang, M. Kosinski, D.J. Stillwell, and J. Rust. Can Well-Being be Measured Using Facebook Status Updates? Validation of Facebookâs Gross National Happiness Index. *Social Indicators Research*, 115(1):483–491, 2014.

Optimizing Sequential Pattern Mining Within Multiple Streams

Daniel Töws, Marwan Hassani, Christian Beecks, Thomas Seidl

Data Management and Data Exploration Group
RWTH Aachen University
Germany

{toews, hassani, beecks, seidl}@cs.rwth-aachen.de

Abstract: Analyzing information is recently becoming much more important than ever, as it is produced massively in every area. In the past years, data streams became more and more important and so were algorithms that can mine hidden patterns out of those non static data bases. Those algorithms can also be used to simulate processes and to find important information step by step. The translation of an English text into German is such a process. Linguists try to find characteristic patterns in this process to better understand it. For this purpose, keystrokes and eye movements during the process are tracked. The *StrPMiner* was designed to mine sequential patterns from this translation data.

One dominant algorithm to find sequential patterns is the PrefixSpan. Though it was created for static data bases, lots of data stream algorithms collect batches and use the algorithm to find sequential patterns. This batch approach is a simple solution, but makes it impossible to find patterns in between two consequent batches. The *PBuilder* is introduced to find sequential patterns with a higher accuracy and is used by the *StrPMiner* to find patterns.

1 Introduction

When translating an English text into German, a lot of cognitive efforts have to be made. The e-cosmos project is a project in the humanities, which tries to find characteristic patterns in human behavior while being confronted with linguistic challenges. As a part of this project test subjects, with German as their native language, were given the task to translate an English text into German. While the subjects translated the text, their eye movement and keystrokes were collected. The linear representation in Figure 1 helps to understand the process and gives small insight into the cognitive process. The pauses marked by the stars, help to find challenges for the subjects but they provide no information on what the subject is doing during this time, aside from not using the keyboard.

To find those hidden information sequential pattern mining algorithms can be used over the resulting data stream. Sequential pattern mining is a special case of the frequent item set mining, where patterns have to be frequent subsequences of the stream. Each pattern

```
wid[*12.672]ersetzt•sich•*******weit********⌫⌫*******⌫⌫⌫⌫
⌫⌫⌫⌫⌫⌫⌫⌫⌫⌫⌫⌫⌫⌫⌫⌫stru⌫äu*bt•sich•gegen•weiteres*•⌫⌫n••Dr⌫
⌫⌫Druck
```

Figure 1: The *Linear Representation* shows part of the translation process. The stars mark pauses in which the subject did not use the keyboard for at least one second.

has to appear a certain number of times within one batch to count as a sequential pattern.

Stream data environments provide data sets that change constantly and provide new information every second. Because of this, each item can only be looked at once and then has to be forgotten.

For this, the algorithm will be split into an online and an offline part. The online part handles the arriving data itself. If the data needs simplification, or preprocessing of any kind, this will be performed by the online part. The detailed analysis will then be done by the offline component. For example some algorithms use the online component to collect data batches, so that the offline component can perform the algorithm for many items in parallel.

Additional challenges arise when looking at multiple streams at once, as patterns can be part of one or multiple streams. In this case the eye movement and keystrokes provide two different streams of data which can be analyzed to help understand the process of text translation. Using the data stream approach to evaluate the information step by step, reconstructs the whole process and gives deep insight for each session.

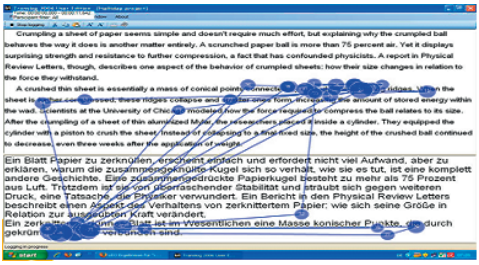


Figure 2: During the experiment eye gazes and keystrokes are recorded. The top half shows the original text and the bottom half the translated version. Fixations are marked in blue.

There are a lot of different algorithms that mine sequential patterns from data streams, but most of them use a batch approach. On the one hand the batch approach is a simple and fast solution to mine sequential patterns in a stream. On the other hand it has a room for errors. To avoid those errors, we will introduce two new algorithms. The PatternBuilder *PBuilder* mines sequential patterns for given data, without using the batch approach. The Streaming Pattern Miner *StrPMiner* uses the *PBuilder* to find sequential patterns in multiple streams and keeps track of their quality.

In section 2, this paper will discuss the related work that has been done in the past. Section 3 looks at the preliminaries of sequential pattern mining. Also it will highlight the problem

with the batch approach. In section 4 two new algorithms are introduced to mine sequential patterns. The *PBuilder* will then be tested and the results will be discussed in section 5. The paper concludes with a summary and a look to future directions in section 6.

2 Related Work

Optimizing sequential pattern mining is an important task in the streaming data mining field which lead to a lot of different algorithms. A base algorithm for many approaches [MDH08],[SEM11],[WC07], [CCPL14] is the *PrefixSpan* algorithm [PHMA+01]. The *PrefixSpan* algorithm was designed for a static data environment. Because of this it can use the A priori assumption that every element contained in a frequent sequential pattern, also has to be frequent. In a given data base, the *PBuilder* will first look for frequent elements which are sequential patterns of length one. Following the A priori assumption, patterns can not be frequent, if they are created with elements or smaller patterns, that are non frequent. Using this, the *PrefixSpan* recursively creates new sequential pattern candidates, by combining a sequential pattern with a frequent element. All new created patterns will be counted and if they are frequent, *PrefixSpan* will search recursively for even longer patterns.. All algorithms using the *PrefixSpan* in a stream environment collect data in a batch instead of evaluating each item alone.

Since the streaming approach allows to only look at data once, algorithms have to make compromises in order to provide fast results. [MDH08] proposes two algorithms with different pruning strategies, the *SS-BE* and *SS-BM* algorithm. These algorithms restrict memory usage but are able to find all true sequential patterns and allow an error bound on the false positives. The patterns are saved in a new designed tree structure, the T_0 tree. Each pattern will only be saved in the tree, if it occurred frequently in the recent time. Both of those algorithms use the *PrefixSpan*.

In a static data set, all information needed for the algorithm are provided from the beginning, while in the streaming approach new data arrives every second, thus, patterns that where not frequent in the beginning may become frequent later on. Yet it is impossible to save every pattern and its information. The *FP-stream* [GHP+03] solves this issue by saving information in different time granularities. The newer the information, the more accurate it will be displayed. Another way to solve the memory problem is by using a sliding window model, in which only the most recent data is being looked at. The *MFI-TransSW* algorithm [LL09] optimizes this concept. The algorithm works in three steps: window initialization, window sliding and pattern generation.

Previously described algorithms only provide solutions for one stream. In cases of multiple streams in parallel, the *MSS-BE* algorithm [HS11] is an idea to find sequential patterns in an multiple stream environment, where pattern elements can be part of different streams.

The algorithms mentioned above either provide solutions for frequent pattern mining or find sequential patterns by using batches.

3 Preliminaries: Optimizing Sequential Pattern Mining

A stream S is defined by an infinite ordered set $S = \{(s_1, t_1), (s_2, t_2), (s_3, t_3), ...\}$, with s_i being the item and t_i the arrival time of the item. A sequential pattern of S is defined as [HS11]:

Given a sequence of items $p = \{a_1, a_2, ..., a_n\}$. The sequence $p' = b_1, b_2, ..., b_m$ is a *subsequence* of p, if there exist integers $i_i < i_2 < ... < i_m$ such that $b_1 = a_{i_1}, b_2 = a_{i_2}, ..., b_m = a_{i_m}$. The *support* of a subsequence p in S is defined as the count of p divided by the number of items that have already arrived in the stream. A subsequence of S is a *sequential pattern* when its support is above a given threshold. In short, a sequential pattern in S is a frequent subsequence in S.

Following the A priori principle, given two subsequences $p = \{p_1, p_2, ..., p_n\}$ and $p' = p \backslash \{p_n\}$, it holds that $supp(p') \geq supp(p)$ due to the anti-monotonicity property. Thus, if p is a sequential pattern, p' is also a sequential pattern. An association rule is an implication of the form $p_1, p_2, ...p_{n-1} \Rightarrow p_n$. The confidence of an association rule is defined as $conf(p_1, ..., p_{n-1} \Rightarrow p_n\} = \frac{supp(p)}{supp(p')}$.

Given multiple streams $S_1, S_2, ..., S_n$, sequential patterns can be generated with either of the streams, or as a mixture of multiple streams. Sequential patterns that only contain items of one specific stream are called intra-stream sequential patterns. Sequential patterns that contain items of different streams are called inter-stream sequential patterns [HS11]. For inter-stream sequential patterns, the rules above have to be adapted. The sequence must not be a subset of one set but a combination of subsets from different sets.

To provide different views on the data, three different window concepts are used by the *StrPMiner*. The algorithm works with the Landmark Window, the Sliding Window and the Damped Window concept. In the Landmark Window, a point in time is defined as the *landmark*. All data is then collected starting from the *landmark*. This concept allows to look at big parts of the data. The Sliding Window concept uses a fixed window size and slides it over the data. Thus, only a snapshot of the data will be monitored at any given time. An advantage is that old patterns will be forgotten eventually, which leaves only current information. The Damped Window uses a similar concept as the Landmark Window. In contrast to the Landmark Window, the Damped Window uses weights to reflect the age of an object. New items will be more important than old ones. This allows a compromise between the Landmark Window and the Sliding Window concept.

3.1 The Batch Approach and its Problems

A good solution to find sequential patterns in a streaming environment is the batch approach. It allows to use the Apriori principle, since each batch provides a static data set. However it also leaves a room for errors.

Given a support threshold of 2, meaning a pattern has to appear two times within one batch

to be counted as frequent, a batch size of 3 and following sequence:

$$(A, B, C, A, C, C, A, D, C, A...) \tag{1}$$

with A, B, C, D being items of a stream. The online component would cut the data stream in following batches:

1. (A, B, C) 2. (A, C, C) 3. (A, D, C) 4. (A, ...) 5. ...

In this case, no pattern would be frequent. Looking at the whole data without cutting it into batches would reveal that the pattern C, A appears three times, which is over the support threshold of 2. Thus, would lead to a frequent pattern. Additionally, all items except for C in the second batch, would be pruned away, although the item A and C appear in every batch. This leads to two reasons for errors through the batch approach:

1. Patterns that appear between batches will not be found.
2. Items and patterns that do not appear often in one batch will be pruned, although they are frequent in the whole data set.

The *StrPMiner* was designed to avoid the batch approach because of these two reasons which result into false statistics for sequential patterns.

4 The StrPMiner

Since the *PrefixSpan* algorithm only scales well when the candidates for sequential patterns can be pruned, the *StrPMiner* reverses the idea of the *PrefixSpan* and uses a new algorithm called the Pattern Builder (*PBuilder*). This allows the *StrPMiner* to work on each data item step by step as it comes in.

For the given data and application the order of the items is important. To provide this focus, the definition of sequential patterns had to be changed slightly. As stated previously, a sequential pattern is a frequent subsequence. A subsequence is, in short, an ordered list of elements taken from the database. Yet for the e-cosmos project, the definition of a subsequence had to be stricter. We redefine a subsequence, and also a sequential pattern, as only allowed to be a list of ordered items which follow each other directly: p is considered a subsequence of q if $p = (p_1, p_2, ..., p_n)$, $q = (q_1, q_2, ..., q_m)$ and there exist integers $i_i < i_2 < ... < i_m$ such that $p_1 = q_{i_1}, p_2 = q_{i_2}, ..., p_m = q_{i_m}$ *and* for all k, l with $l, k < m$ and $l = k + 1$ there exists no item q_j with $k < j < l$. This restriction was made, since the focus of the patterns should lie in the order of the items.

The *StrPMiner* handles arriving data from multiple streams at once, compresses the data and passes it to the *PBuilder*. The *PBuilder* then uses this data to create sequential pattern candidates. After this, the *StrPMiner* saves the candidates in the T_0 tree structure and keeps track of those candidates and their corresponding statistics. Currently those are the count, support and confidence value. This approach allows full accuracy, and flexibility

in the output, as the support threshold can be changed at every output request. This is not possible when using the *PrefixSpan*, as the threshold has to be set before calculation starts. Additionally the *StrPMiner* is able to jump at any point in time of the calculation process.

4.1 The *PBuilder*

The *StrPMiner* assumes that at each point in time, only one item can arrive. If more than one item arrives at one point in time, the algorithm orders them and calculates each step for one item after another. This allows the *StrPMiner* to minimize the amount of calculation that has to be done with each new arriving item by only updating the statistics that are affected by the new item.

Following this concept, the *PBuilder* is only creating patterns that contain the newly arrived item. Additionally, since it is the last arrived item, all created patterns will end with this item. Given an item A as the newly arrived item, the *PBuilder* starts with this item as the first pattern, the postfix. After this, the algorithm recursively adds older items as a prefix to the previously created postfix. To ensure that the *StrPMiner* only finds direct sequential patterns, the prefix is a direct predecessor of the postfix Figure 4 visualizes the *PBuilder*. Also Algorithm 1 shows a peseudocode of the *PBuilder*

The *PBuilder* saves the relevant items in a list. The list contains all relevant items, without differing between the streams. With this concept, the *PBuilder* can create all inter- and intra-stream patterns without any specifications.

For each created pattern, the *PBuilder* algorithm calls the update function of the T_0 tree.

```
PBuilder
Data: DataSnapshot, currentPattern
//DataSnapshot contains the latest compressed items and is limited by maxRuleLength.
The newly arrived item is at the last position
Result: The new patterns that can be created with the new item
int index = DataSnapshot.length;
//create patterns until maxRuleLength is reached
while currentPattern.length ≤ DataSnapshot.length do
    //add the next item to the pattern
    currentPattern = DataSnapshot.get(index-currentPattern.length) + currentPattern;
    //update the tree with the new pattern
    updateTree(currentPattern);
end
```

Algorithm 1: The *PBuilder* explained with pseudo code

4.2 Maximum Pattern Length as a Solution for Exponential Growth

In contrast to a static database, where all information are available from the beginning, the streaming approach does not have any information on how future items and their frequency might look like. This means that any item and pattern that is currently not frequent in a stream, can become frequent at any later point in time. The support of every pattern changes with every new arriving item. To ensure that at every time the user requests an output all sequential patterns are part of the output, every possible pattern and its information have to be saved. This causes an exponential increase of the calculation time, as with every new arriving item more patterns can be created. Additionally, the memory space will eventually collapse, as the amount of data that has to be saved also increases exponentially.

To stop the exponential growth, the *StrPMiner* introduces a parameter called *maxPatternLength*, as an upper Bound for the pattern length. This variable restricts the *PBuilder* to only look at the last *maxPatternLength* items. A *maxPatternLength* of five, will cause patterns to maximally contain five items, as only those are relevant to the algorithm. Given this bound, the calculation time in each step only scales with the size of the *maxPatternLength* parameter. Additionally this parameter bounds the maximum growth of the required memory space. On the one hand, as the parameter will not change over the time, the calculation time for each new arriving item will be constant. On the other hand this upper bound filters patterns, before they have been created. Sequential patterns that have a length higher than the given bound, will not be found. With this in mind, a careful selection of the upper bound is important, as it provides a trade off between the calculation time and accuracy.

4.3 Different Window Models

As previously mentioned, the *StrPMiner* uses the T_0 tree introduced by [MDH08]. For the algorithm slight adaptations were made, regarding the saved information. The *StrPMiner* saves the label of the item and the appearances of the pattern in each node. The count of each pattern is then determined by the number of time stamps saved in the corresponding node. An example is shown in Figure 5.

The sliding window model helps to provide another view on the data, as it only contains knowledge of recent data and forgets old data. This helps in cases, where the data changes drastically over the duration of the stream. The landmark window would still show old patterns even though they did not reappear for a long time. In general the whole algorithm works the same, as in the landmark window, except for an extra pruning step. For this the time stamp of the corresponding item and the patterns created with it have to be deleted from the T_0 tree, which is one path.

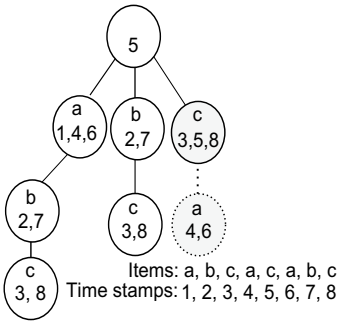
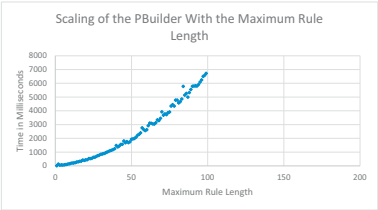
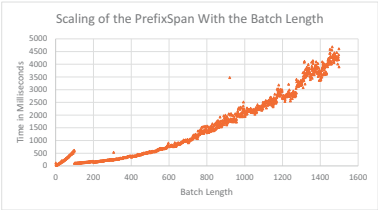


Figure 3: An example of the T_0 tree. The doted node represents the pattern (c,a).



(a) The *PBuilder* scales with the Maximum rule length parameter



(b) The *PrefixSpan* scales with the batch length

Figure 4:

5 Experimental Results

Because of the two reasons previously described, the *StrPMiner* uses a different algorithm to find sequential patterns, than the *PrefixSpan*. The *PBuilder* can handle each newly arriving item without using the batch approach.

This results in an algorithm that uses more memory space, as it has to save all patterns. While the calculation time of the *PrefixSpan* scales with the set support threshold and the batch length, the calculation time of the *PBuilder* is only dependent on the maximum pattern length.

For this experiment the *PrefixSpan* has been adapted, so that it will only create direct sequential patterns. Additionally this improves the run time of the *PrefixSpan*, as it will calculate less patterns.

As seen in Figure 6 the calculation time of both algorithms increases exponentially with the increase of their corresponding parameters. The scaling of the *PBuilder* may be worse, but in practical use this is no real problem. Rarely sequential patterns above a threshold of 1% have a length of more than 20. In the given translation data set, there are no patterns

with a length of more than 18, that have support over 1%. With the assumption that no sequential pattern has a length of more than 20, the *PBuilder* provides a 100% accuracy with a maximum pattern length parameter of 20.

	Datasize	Batchlength 50		Batchlength 250		Batchlength 1000	
Subject		max. error	avg. error	max. err	avg. err	max. err	avg. err
GT1	3885	57/498	35/545	50/406	23/545	89/406	27/545
GT2	3926	59/194	30/371	65/222	38/371	132/233	42/371
GT3	3913	53/455	30/473	47/298	27/473	78/455	17/473
GT4	5503	101/322	68/814	51/273	42/814	75/369	33/814

Table 1: The table shows the error rate on the count value over the Top 20 patterns with the highest support value, generated by the *PrefixSpan* and the *PBuilder*. The support threshold was set to 0.05. The notation x/y reads as, y being the number of patterns found by the *PBuilder* and x the difference between y and the number of patterns found by the *PrefixSpan*.

Table 1 and Table 2 show the error rate of the *PrefixSpan*. The 20 patterns with the highest support value of each algorithm were counted and compared. The ranking of the patterns only has minor differences. The accuracy of the *PrefixSpan* should increase with the size of the batch length less patterns can be lost between two batches. Nevertheless, Table 1 shows, that the batch length may be selected unluckily, so that some sequential patterns will be pruned because they are part of different batches. This may also provide mistakes in the count value of items, as an item may not be frequent in one batch, although it is frequent overall.

	Datasize	Batchlength 50		Batchlength 250		Batchlength 1000	
Subject		max. error	avg. error	max. err	avg. err	max. err	avg. err
GT1	3885	39/367	21/545	13/211	5/545	10/211	2/545
GT2	3926	17/194	6/371	5/212	2/371	1/395	0.1/371
GT3	3913	28/361	14/473	8/361	3/473	13/213	2/473
GT4	5503	90/564	47/814	29/564	15/814	14/273	5/814

Table 2: Error rate as in Table 1. Here the support threshold is set to 0.01.

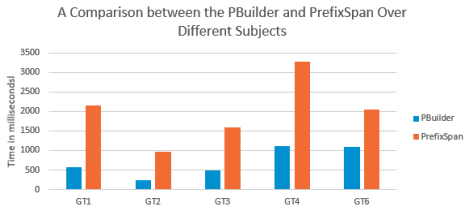


Figure 5: A visualization of the calculation time of both algorithms on different data sets. The maximum rule length is set to 20 and the batch length to 1000. The support threshold is 0.01.

Table 2 shows, that with the proper settings, the errors made by the *PrefixSpan* are neglectable. But using those settings increases the calculation time to such extent, that the *PBuilder* is much faster. This is visualized in Figure 7.

6 Conclusion and Future Directions

The *PBuilder* is an algorithm to mine sequential patterns out of a streaming data environment. Experiments have shown, that the calculation time of the *PBuilder* may be slower than the calculation time of the *PrefixSpan*, but it is still good enough. Additionally the *PBuilder* has a higher accuracy rate, which shows, that the *PBuilder* provides a good alternative to the *PrefixSpan*. The *StrPMiner* uses the *PBuilder*, which allows it to be more flexible than most current algorithms.

Fixations made by the eyes have a temporal extension. This means they start at a specific point in time and end at another, later, point in time. Those so called interval-based events and their temporal relations can not be found by the *StrPMiner*. In the future, we will upgrade the *StrPMiner* to the Interval Streaming Pattern Miner (*IStrPMiner*). The *IStrPMiner* will be able to find characteristics in the temporal relationships between different items. Those patterns are of the form: A is overlapping with B, or A happens during B.

Acknowledgments

Funded by the Excellence Initiative of the German federal and state governments.

References

[CCPL14] Yi-Cheng Chen, Chien-Chih Chen, Wen-Chih Peng, and Wang-Chien Lee. Mining Correlation Patterns among Appliances in Smart Home Environment. In *Advances in Knowledge Discovery and Data Mining*, pages 222–233. Springer, 2014.

[GHP+03] Chris Giannella, Jiawei Han, Jian Pei, Xifeng Yan, and Philip S Yu. Mining frequent patterns in data streams at multiple time granularities. *Next generation data mining*, 212:191–212, 2003.

[HS11] Marwan Hassani and Thomas Seidl. Towards a mobile health context prediction: Sequential pattern mining in multiple streams. In *MDM, 2011*, pages 55–57. IEEE, 2011.

[LL09] Hua-Fu Li and Suh-Yin Lee. Mining frequent itemsets over data streams using efficient window sliding techniques. *Expert Systems with Applications*, 36(2):1466–1477, 2009.

[MDH08] Luiz F Mendes, Bolin Ding, and Jiawei Han. Stream sequential pattern mining with precise error bounds. In *ICDM.*, pages 941–946. IEEE, 2008.

[PHMA+01] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *ICDE*, pages 0215–0215. IEEE Computer Society, 2001.

[SEM11] Amany F Soliman, Gamal A Ebrahim, and Hoda K Mohammed. SPEDS: A framework for mining sequential patterns in evolving data streams. In *Communications, Computers and Signal Processing (PacRim), 2011*, pages 464–469. IEEE, 2011.

[WC07] Shin-Yi Wu and Yen-Liang Chen. Mining nonambiguous temporal patterns for interval-based events. *KDE*, pages 742–758, 2007.

Konzept und prototypische Implementierung eines föderativen Complex Event Processing Systems mit Operatorverteilung

Marcus Pinnecke

Fachbereich Mathematik und Informatik
Philipps Universität Marburg*

Abstract: Complex Event Processing (CEP) ist eine etablierte Technologie zur Verarbeitung von Ereignisströmen in nahezu Echtzeit. Trotz alledem unterscheiden bestehende CEP-Systeme sich stark in ihren jeweiligen Leistungsumfängen und -profilen. Werden verschiedene CEP-Systeme in einem föderativen System kombiniert, so kann das resultierende System im Vergleich zu den Einzelsystemen eine höhere Datendurchsatzrate und einen breiteren Leistungsumfang erreichen. Fehlende Standardisierung und inkompatible Schnittstellen der CEP-Systeme behindern diesen Ansatz jedoch. Dieser Missstand wird durch die Middleware Java Event Processing Connectivity (JEPC) behoben. Gegenstand dieser Arbeit ist das auf JEPC basierendes föderatives CEP-System so zu erweitern, dass eine Verteilung von Anfrageoperatoren auf die beteiligten Systeme automatisch durchführt wird. Hierfür wird in dieser Arbeit das zugrundeliegende Konzept, eine Anfrageoptimierung sowie ein Kostenmodell für die Auswahl eines konkreten Systems vorgestellt. In einer Evaluation wird gezeigt, dass durch den Einsatz einer Föderation die Datendurchsatzrate deutlich steigt, insbesondere wenn es sich um eine heterogene Föderation mit unterschiedlichen Leistungsprofilen handelt.

1 Einleitung und Motivation

Complex Event Processing (CEP) ermöglicht kontinuierliche Analysen, Prognosen und Reaktionen basierend auf Veränderungen in Ereignisströmen [Luc01]. Währenddessen klassische Datenbanksysteme auf einer persistenten Datenbasis arbeiten, werden in CEP-Systemen *kontinuierliche Anfragen* auf einen Strom von Ereignissen ausgeführt, die als Ergebnis wieder einen Ereignisstrom liefern [BW01]. Beim Leistungsumfang und Datendurchsatz unterscheiden sich CEP-Systeme verschiedener Hersteller erheblich voneinander.

Um eine geringe Latenz oder einen hohen Durchsatz zu einer vorgegebenen Latenz zu erreichen, könnte ein Mix aus Systemen in Form einer Föderation in Betracht gezogen werden. Ein Ansatz ist föderative Datenbanksysteme auf das CEP-Szenario anzupassen. Hierzu wird das föderative Datenbanksystem um kontinuierliche Anfragen erweitert [ea09]. Ein anderer Ansatz ist eine reine Föderation aus CEP-Systemen anzubieten. Dieser Ansatz

*Seit WS 2014/2015 für Informatik M.Sc. Studium: Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg. E-Mail: marcus.pinnecke@ovgu.de

ist Grundlage dieser Arbeit und ermöglicht eine wesentlich feinere Verteilung von kontinuerlichen Anfragen, indem diese operatorweise auf beteiligte Systeme (den *Pool*) verteilt werden. Hierdurch wird eine kollaborative Verarbeitung der Anfrage im Pool möglich. Die Zusammenstellung des Pools nimmt maßgeblich Einfluss auf das Leistungsprofil des föderativen Systems. Besteht der Pool aus einer Menge von Systemen des gleichen Herstellers (*homogene* Föderation), gibt es zwar keine Schnittstellenprobleme zwischen den Systemen, aber das föderative System ist auf den Leistungsumfang des gewählten Systems beschränkt. Werden Systeme unterschiedlicher Hersteller (*heterogene* Föderation) genutzt, so kann der Leistungsumfang im Vergleich zu einer homogenen Föderation gesteigert werden. Möglich wird dies durch die Kompensation fehlender Funktionalität eines Systems durch ein anderes System. Weiterhin kann für eine Operation ein Spezialist aus dem Pool gewählt werden, der diese besonders effizient bearbeiten kann (*Leistungsprofil*). Um neben einem hohen Datendurchsatz zusätzlich einen großen Leistungsumfang für die Föderation zu erreichen, muss ein heterogener Pool genutzt werden.

Die Heterogenität des Pools führt sofort zu einem Kommunikationproblem der Systeme untereinander, da es keine durchgehend implementierten Standards für Systeme unterschiedlicher Hersteller gibt (vlg. [JMS+08]).

Die Middleware *Java Event Processing Connectivity* (JEPC) begegnet dieser Herausforderung durch die Einführung einer von den herstellerspezifischen Details abstrahierten Beschreibungssprache für Ereignisströmen und Anfragen. Anweisungen in dieser Sprache werden durch JEPC vollautomatisch in die jeweilige Systemsprache übersetzt. Werden Anwendungssysteme (AWP) gegen die Schnittstelle von JEPC entwickelt, so kann das zugrundeliegende CEP-System mühelos durch ein anderes ersetzt werden [HGMR13].

Motiviert durch den resultierenden Nutzen oder oben genannten Ideen, ist der Ansatz dieser Arbeit die Weiterentwicklung des auf JEPC basierenden föderativen CEP-Systems, so dass dieses automatisch auf einen heterogenen Pool zurückgreifen kann. Ein AWP interagiert über die Schnittstelle von JEPC mit diesem föderativen System, wobei die konkrete Verarbeitung an den Pool delegiert wird. Diese Aufgabenübertragung erfolgt durch eine Verteilung der Anfrageoperatoren auf die Systeme innerhalb des Pools und führt zu einer Arbeitsteilung, indem eine Anfrage kollaborativ durch verschiedene Systeme prozessiert werden kann. Bei der Verteilung der Operatoren werden die Leistungsprofile der jeweiligen Systeme berücksichtigt, so dass dies zu einer Beschleunigung der Anfrageverarbeitung insgesamt führt. Ein geeignetes Kostenmodell für die konkrete Auswahl eines Systems führt weiterhin zu einer gleichmäßigeren Lastverteilung zwischen den Systemen, so dass vorhandene Ressourcen effizienter genutzt werden können.

Im Folgenden wird ein Überblick der Architektur des föderativen CEP-Systems im Abschnitt 2 vorgestellt. Dieser Überblick stellt die Aufgaben des Pools, des föderativen Systems und der AWP zueinander in Beziehung. Die Berücksichtigung der Leistungsprofile bei der Verteilung eines Operators auf ein System basiert auf einem heuristischen Regelwerk, das in Abschnitt 3 ausführlicher beschrieben wird. Eine Evaluation des Systems findet darauffolgend im Abschnitt 4 statt. Hier wird das föderative Systeme einmal mit einem homogenen und einmal mit einem heterogenen Pool in zwei Szenarien untersucht. Nach der Betrachtung verwandter Arbeiten im Abschnitt 5, bildet der Abschnitt 6 den Abschluss. In diesem wird eine Zusammenfassung und ein Ausblick auf die zukünftige

Ausrichtung des Projekts gegeben.

2 Architektur

Die Architektur des föderierten CEP-Systems lässt sich schematisch wie folgt einteilen. Auf der einen Seite befinden sich AWP, die beliebige kontinuierliche Anfragen an das föderative System stellen. Hierbei kommunizieren die AWP mit den durch JEPC bereitgestellten Schnittstellen des föderierten Systems. Auf der anderen Seite ist ein Pool bestehend aus verschiedenen Event-Service-Providern (ESP)[1], die ebenfalls durch JEPC an das föderierte System angebunden sind. Im Rahmen des Prototyps wurden folgende ESP von JEPC verwendet:

- **N** die Implementierung von CEP-Funktionalität direkt in JEPC,
- **E** ein Adapter für das CEP-System *Esper*,
- **C** ein Adapter zu einem kommerziellen CEP-System und
- **D** eine Emulation von CEP-Funktionalität in Datenbanken über JDBC.

Notation: Dem besseren Lesefluss bedingt, wird ein föderiertes System bestehend aus den Systemen A und B im Folgenden durch AB abgekürzt. So beschreibt zum Beispiel *NECD* eine Föderation aus allen vier oben genannten ESP und *EE* eine Föderation aus zwei Esper ESP.

2.1 Zentrale Komponenten

Zwischen den AWP und dem Pool befindet sich das föderierte System, das die drei folgenden Komponenten umfasst:

Anfragegraph. Der globale Anfragegraph enthält alle durch die AWP an die Föderation gestellten Anfragen. Er ist für den Transfer der Daten zwischen den beteiligten Systemen untereinander, zur Föderation und zu den AWP zuständig. Dieser Graph umfasst Metainformationen über die Verteilung, um eine günstige Weiterleitung der Ergebnisse zwischen den Systemen zu organisieren. Teilanfragen, die innerhalb des zugewiesenen Systems verarbeitet werden können, werden autonom und unabhängig innerhalb des ESP prozessiert (Autonomie-Maximierung). Die kontinuierlichen Ergebnisse der ESP werden durch das föderative System abgefangen und an die AWP sowie abonnierenden ESP weitergeleitet. Die Wahl der Graphenstruktur als Basis für eine Föderation ist maßgeblich durch das Ereignisstromparadigma geprägt, das Anfragen als loses gekoppeltes Netz von Operatoren beschreibt. Die Übernahme dieser Struktur ermöglicht eine feingranulare Aufteilung einer Anfrage, indem die Operatoren der Anfrage verteilt werden.

Klassifikationsverfahren. Das Klassifikationsverfahren schlägt für jeden Operatortypen (Filter, Aggregator, Korrelator, Join und Mustererkennung) eine Menge von homogenen Systemen innerhalb des Pools vor (*Zielmenge*), deren Wahl günstig für den gegebenen Operator ist. Die Wahl eines konkreten Systems aus der Zielmenge wird allerdings durch die *Systemzuweisung* ausgeführt und berücksichtigt weitere Kriterien, wie z.B. die aktuelle Auslastung der Systeme innerhalb der Zielmenge. Der vom Klassifikationsverfahren gegebene Vorschlag berücksichtigt sowohl das Leistungsprofil als auch den Leistungsumfang

[1] Ein konkretes CEP-System oder eine Emulation innerhalb einer Datenbank [HGMR13]

und versucht beides zu optimieren. Im Rahmen des Prototyps ist das Klassifikationsverfahren durch einen Entscheidungsbaum umgesetzt, welcher basierend auf einer Heuristik schnell gute Platzierungen ermittelt und eine sinnvolle initiale Verteilung der Operatoren durchführt.

Systemzuweisung. Nachdem für jeden Operator einer Anfrage eine Zielmenge bestimmt ist, wird jeder Operator einzeln auf ein konkretes System (*Zielsystem*) seiner Zielmenge bereitgestellt. Die Wahl des Zielsystems wird durch ein Kostenmodell bestimmt, dass die momentane Systemauslastung aller Systeme innerhalb der Zielmenge berücksichtigt und eine möglichst gleichmäßige Lastverteilung innerhalb der Zielmenge anstrebt.

2.2 Verwalten von kontinuierlichen Anfragen

Aus der Perspektive eines AWP verhält sich das föderierte System wie jedes andere System, das der Spezifikation von JEPC entspricht. Währenddessen JEPC eine Anfrage direkt in die herstellerspezifische Sprache übersetzen würde, werden innerhalb des föderierten Systems folgende Schritte durchgeführt: *(1)* Die Anfrage wird in den globalen Anfragegraphen übernommen, *(2)* im Anschluss wird für jeden noch nicht zugewiesenen Operator durch die Klassifikation ein prädestiniertes System gewählt, *(3)* jeder Operator wird anhand der Systemzuweisung in Abhängigkeit des Kostenmodells auf ein konkretes System im Pool übertragen. Dieser Vorgang ist in Abbildung 1 dargestellt. Bei diesem Vorgang werden die betroffene ESP im Pool über ihre JEPC Schnittstelle angesprochen, so dass diese Übertragung zu einer Übersetzung der (Teil-)Anfrage in die herstellerspezifische Sprache führt. Aus Sicht eines ESP ist die ihm übertragene (Teil-)Anfrage eine vollständige kontinuierliche Anfrage, die wie gewohnt verarbeitet werden kann. Für einen Übergang

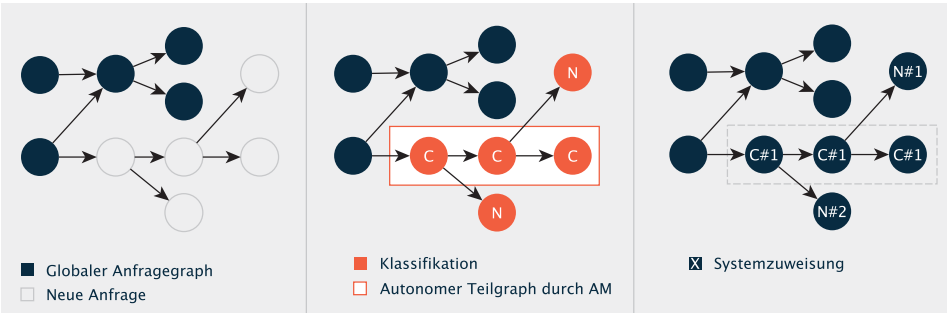


Abbildung 1: Dem globalen Anfragegraphen wird eine neue Anfrage hinzugefügt. Jeder Operator wird klassifiziert und anschließend auf ein Zielsystem übertragen. In der Abbildung werden zwei Operatoren für dieselbe Zielmenge *N* an zwei unterschiedliche Zielsysteme *N#1* und *N#2* delegiert. In der mittleren Abbildung ist die Autonomie-Maximierung (AM) dargestellt, die einen Teilgraph komplett überträgt, so dass für alle Operatoren dieses Teilgraphen das Zielsystem *C#1* der Zielmenge *C* identisch ist.

von einem System zu einem anderen System fängt das föderative System die Ausgaben vom ersten System ab und leitet diese an das zweite System weiter (*Inter-System-Kommunikation*). Bei einer naiven Verteilung der Operatoren kann es zu einer unnötigen Kommunikation zwischen der Föderation und den Pool kommen. Damit dies verhindert

wird, wird die Autonomie-Maximierung durchgeführt.

Autonomie-Maximierung (AM). Ohne weitere Optimierung werden die Ergebnisse *jedes* Operators durch das föderierte System abgefangen und weitergeleitet, da jeder Operator einzeln auf einem Zielsystem bereitgestellt wird. Dies tritt auch dann ein, wenn Teilanfragen durch dasselbe Zielsystem prozessiert werden und keine Kommunikation mit dem föderierten System für die Verarbeitung dieser Teilanfrage nötig ist. Diese Indirektion ist Folge des einheitlichen Aufbaus der Inter-System-Kommunikation und führt in Grenzfällen zu einem vermeidbaren Kommunikationsaufwand. Die AM ist eine optionale Optimierungsstrategie, die möglichst große Teilanfragen einer kontinuierlichen Anfrage *gemeinsam* für ein Zielsystem vorsieht, anstelle dies *operatorweise* zu tun, um Kommunikationskosten zwischen dem Pool und dem föderierten System einzusparen. Standardmäßig ist dieses Verfahren aktiviert.

> **AM-Kriterium:** Die AM kann auf jeder Teilanfrage einer kontinuierlichen Anfrage, deren Operatoren *(1)* direkt miteinander verbunden und *(2)* nach der Anwendung der Klassifikation für dieselbe Zielmenge vorgesehen sind, ausgeführt werden.

In Folge der AM können ESP möglichst große Teilanfragen selbständig bearbeiten, so dass der Aufwand für die Kommunikation mit dem föderierten System eingespart werden kann. Diese Herangehensweise zeigt positive Effekte in Bezug auf die Datendurchsatzrate in heterogenen Föderationen, wie in der Evaluation in Abschnitt 4 zu sehen ist. Die hierfür nötigen Informationen werden im globalen Anfragegraphen hinterlegt.

Ein unerwünschter Nebeneffekt ist die Einschränkung der kollaborativen Verarbeitung von Anfragen. Hierzu soll folgendes Beispiel betrachtet werden: Nach Anwendung der AM auf eine beliebige Anfrage Q an eine homogene Föderation A_1A_2 wird Q komplett an A_1 delegiert ohne A_2 zu berücksichtigen, da *ganz* Q das AM-Kriterium erfüllt und somit vollständig an ein einzelnes Zielsystem übergeben wird - obwohl eine höhere Verarbeitungsgeschwindigkeit durch eine bessere Lastverteilung zwischen A_1 und A_2 im Fall einer gemeinsamen Verarbeitung von Q möglich sein könnte. Welchen Einfluss dieser Nebeneffekt hat und ab welchem Verhältnis zwischen kollaborativer Abarbeitung und Autonomie innerhalb einer homogenen Systemmenge die AM automatisch deaktiviert werden sollte, ist zum aktuellen Zeitpunkt noch offen.

Zuweisung zu einer Systeminstanz. Da einzelne Ereignisströme unterschiedlich stark frequentiert und Operatoren unterschiedlich rechenintensiv sind, kann es zu einer Schieflage der Auslastung in der Zielmenge für einen Operator kommen. Um dies zu vermeiden wird eine gute Balance der Systeme durch folgendes Kostenmodell angestrebt: Zunächst wird die momentane Auslastung jedes Systems als Kriterium herangezogen. Findet sich ein eindeutiges kleinstes Element, so ist ein System mit größtmöglicher freier Kapazität gefunden. Das Verfahren wählt dieses System und ist beendet. Sollte jedoch kein eindeutiger Kandidat existieren, so wird aus der Menge der Kandidaten jenes gewählt, das in vorangegangenen Systemzuweisungen am wenigsten berücksichtigt wurde. Sollte dies eindeutig sein, kann ein System gewählt werden und das Verfahren ist beendet. Wurde das Verfahren nicht beendet, so wird zufällig eine Instanz aus der Kandidatenmenge gewählt und beendet.

3 Ermitteln der Zielmenge pro Operator

Eine optimale Verteilung minimiert die Kommunikationskosten und maximiert den Datendurchsatz über alle Systeme. Dies durch Berechnung zu ermitteln, kann jedoch nicht erwartet werden, da es sich um ein komplexes kombinatorisches Problem handelt. Der für diese Arbeit gewählte Ansatz ist eine Heuristik, die eine gute Startplazierung der Operatoren ermittelt. In einer zukünftigen Entwicklung könnte diese Plazierung durch *dynamische Planmigration* weiter verbessert werden [ZRH04]. Das Verfahren zur Startplazierung eines Operators ist durch den in Abbildung 2 dargestellten Entscheidungsbaum beschrieben. Dieser Entscheidungsbaum bestimmt für einen Operatortypen eine günstige

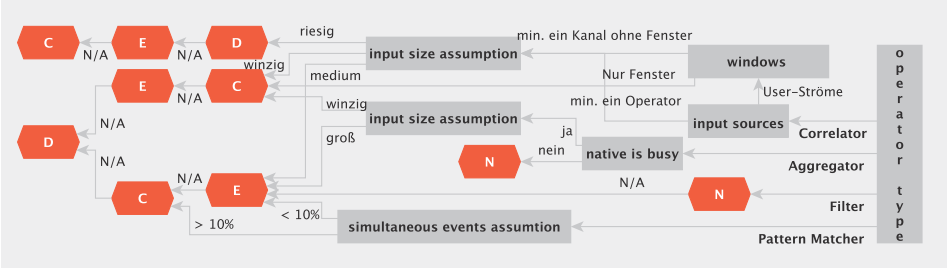


Abbildung 2: Vollständiges Regelwerk zur Bestimmung eines Systems. Eine Kante stellt eine geforderte Merkmalausprägung für die Testfunktion dar. Erfüllt eine Ausprägung dieses Kriterium, wird mit der nächsten Testfunktion fortgefahren. Durch rekursive Anwendung wird eine Zielmenge ermittelt.

Zielmenge, indem eine Reihe von Merkmalausprägungen geprüft werden. Soll beispielsweise ein Filter-Operator platziert werden, so wird zunächst die Implementierung von CEP-Funktionalität direkt in JEPC (N) vorgeschlagen. Sollte dieser ESP nicht im Pool vorhanden oder ausgelastet sein, wird die bestmögliche Alternative gewählt (Esper, kommerzielle ESP oder als letztes JDBC). Neben dem Operatortypen müssen Annahmen zur Statusgröße[2] des Operators *Aggreagtor* und des *Joins* sowie eine Einschätzung zur relativen Anzahl gleichzeitig eintretender Ereignisse[3] gegeben werden. Diese Informationen werden in Abhängigkeit des zu klassifizierenden Operators genutzt, um im Entscheidungsbaum eine geeignete Verzweigung zu wählen.

Die Güte des Entscheidungsbaums wird durch seine Regeln bestimmt. Um gute Regeln zu finden, wurden Benchmarks zu den verfügbaren ESP in JEPC durchgeführt. Durch diese Benchmarks wurden die durchschnittlichen Durchsatzraten der ESP in unterschiedlichen Szenarien je Operatortyp ermittelt. Diese Szenarien untersuchten den Einfluss von Parametern, wie dem Selektivitätsgrad des Filters, der Komplexität von Join-Bedingungen oder dem Umfang der Nutzlast. Im Anschluss wurden die Leistungsprofile anhand markanter Eigenschaften isoliert.

[2] Die Statusgröße bezieht sich auf die Anzahl vorgehaltenen Tupel für statusbehaftete Operatoren.
[3] Dies betrifft Ereignisse mit gleichen Zeitstempel.

4 Experimente

In diesem Abschnitt wird untersucht, welches Potential ein föderiertes CEP-System mit einem heterogenen gegenüber einem homogenen Pool besitzt und welchen Einfluss die AM auf heterogene Föderationen nimmt. Hierbei wird der Durchsatz (Tupel pro Sekunde) der jeweiligen Föderationen ermittelt und gegenübergestellt. Dabei wächst mit laufenden n die Anfragekomplexität (Anzahl der Operatoren).

Setup. Die Experimente wurden jeweils mit 2 Millionen Tupeln zu je 28 Byte auf einem virtualisierten Server mit 8 Intel^{R} Core^{TM}2 Duo-CPU T7700@2.40GHz mit 32 GB RAM durchgeführt, wobei die CPUs exklusiv für die VM reserviert waren. Da die Berechnungen auf den CPUs ausgeführt wurden, die keiner anderen VM zur Verfügung standen, ist eine Verzerrung durch die Virtualisierung nicht zu befürchten. Das Betriebssystem war Ubuntu 14.04 LTS. Jeder Durchlauf $1, \ldots, n$ wurde pro Föderation in einem separaten Prozess ausgeführt. Die Gesamtausführung erfolgte sequentiell.

Zunächst wurde der Durchsatz eines einzelnes ESP gegenüber einer Föderation bestehend aus nur diesem einzelnen ESP ermittelt. Wie zu erwarten war, rentiert sich der Aufwand für die zusätzliche Föderationsschicht nicht, wenn der Pool nur aus einem einzelnen System besteht, obwohl die Leistungseinbuße gleichfalls vernachlässigt werden kann. Demgegenüber zeigte ein Vergleich zwischen einem einzelnen ESP und einer heterogenen Föderation, dass die Föderation in den meisten Fällen deutlich höhere Durchsatzraten erzielt und in keinem der untersuchten Fälle signifikant schlechter war als das einzelne System. Motiviert hierdurch, wurde nun der Einfluss der AM für homogene und heterogene Föderationen untersucht.

4.1 Größtmögliche Autonomie-Maximierung

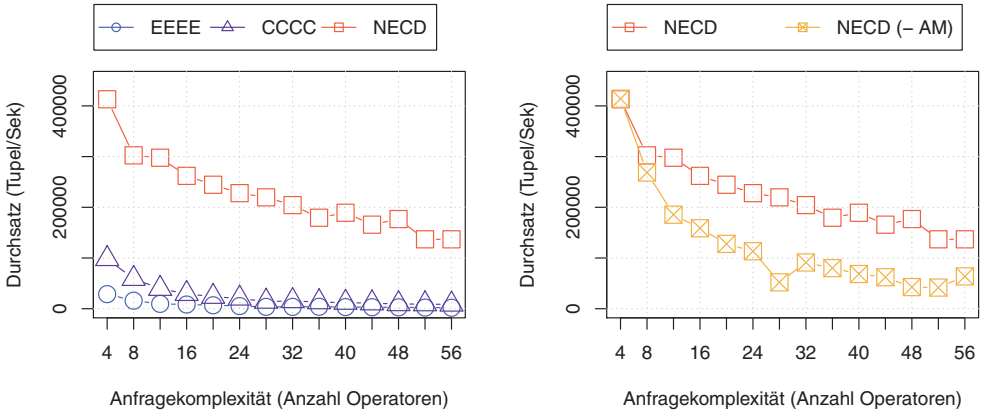


Abbildung 3: Durchsatzraten in Abhängigkeit der Anfragekomplexität (Anfrage q_1)

Eine optimale Anfrage für die AM wird durch die erste Anfrage q_1 erreicht. Diese Anfrage umfasst eine Reihenschaltung von Operatorgruppen. Jede dieser Gruppen ist so organisiert, dass sie nur Operatoren vom gleichen Typ umfasst. Für jeden Operatortyp gibt

es genau eine Gruppe. Hierdurch sind Operatoren gleichen Typs unmittelbar verbunden und können als Ganzes an das bestmögliche System übertragen werden. Es bestehen keine Abhängigkeiten der Gruppen untereinander, außer beim ersten bzw. letzten Operator jeder Gruppe.

Ergebnisse. Die Ergebnisse sind in Abbildung 3 dargestellt und zeigen einerseits einen deutlich erhöhten Datendurchsatz der heterogenen Föderation (NECD) gegenüber den homogenen Föderationen (EEEE bzw. CCCC). Dies wird durch die nahezu optimale Plazierung der Operatoren auf den jeweils am besten geeigneten Systemen möglich. Gemäß dem Entscheidungsbaum (Abbildung 2) und der AM wird jede Operatortyp-Gruppe auf genau ein System der Föderation abgebildet. Da einer homogenen Föderation nur gleichartige Systeme zur Verfügung stehen, können keine Spezialisten für einen bestimmten Operatoren gewählt werden, währenddessen dies in einer heterogenen Föderation möglich ist. Letzteres erklärt die erhöhte Leistung der heterogenen Föderation. Aufgrund der Reihenschaltung der Operatortypen und der Definition der AM bleibt die Anzahl potentiell autonomer Einheiten trotz wachsendem n konstant. Es wird deutlich, dass die AM eine Leistungssteigerung bringt.

4.2 Kleinstmögliche Autonomie-Maximierung

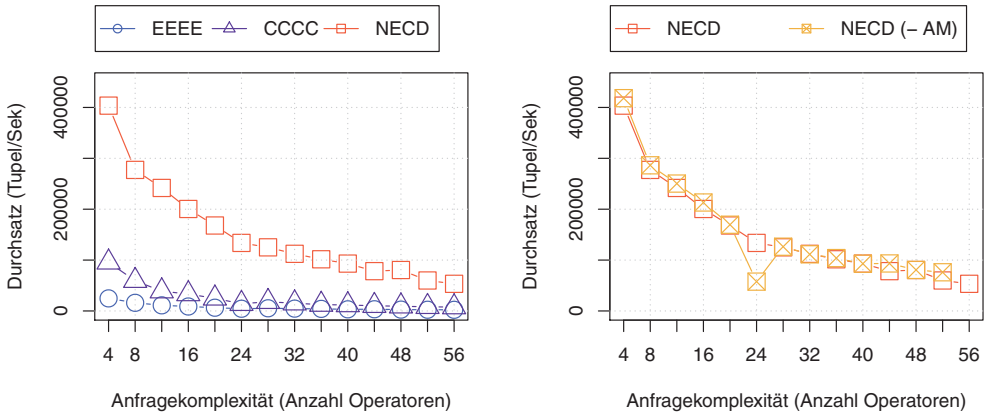


Abbildung 4: Durchsatzraten in Abhängigkeit der Anfragekomplexität (Anfrage q_2)

Eine möglichst ungünstige Struktur für die AM wird in q_2 erreicht, indem keine zwei Operatoren gleichen Typs miteinander verbunden sind. Somit ist q_2 das Gegenstück zu q_1. Die Anfrage q_2 muss *operatorweise* auf den ESP-Pool verteilt werden. Damit kann die AM keine Teilgraphen finden, die autonom an die ESP delegiert werden können.

Ergebnisse. Wie in q_1 wird der Vorteil der heterogenen Föderation deutlich. Die Ergebnisse sind in Abbildung 4 dargestellt. In diesem Szenario können die Operatoren von q_2 nur einzeln an die jeweiligen ESP übergeben werden. Hierdurch steigt der Interaktionsaufwand mit der Föderation im Vergleich zu q_1. Produziert ein Operator ein Ergebnis, muss dieses durch den Föderationsdienst abgefangen und an den nachfolgenden Operator weitergeleitet werden. Da direkt verbundene Operatoren auf unterschiedlichen Systemen

ausgeführt werden, ist diese Indirektion nötig. Dies ist eine Konsequenz aus der Struktur von q_2 und dem Ablauf der AM. Indem zwei verbundene Operatoren nicht vom gleichen Typ sind, wird jeder Operator zwangsläufig auf ein anderes System gegeben. Für direkt verbundene Operatoren bedeutet dies zwangsweise eine erhöhte Kommunikation. Dies zeigt sich in der stärker abfallenden Leistungskurve aller Systeme. Dass die AM in diesem Szenario keine zusätzlichen Vorteile bringen kann, ist den Ergebnissen ebenfalls zu entnehmen. Mit Ausnahme eines Ausreißers unterscheiden sich die Durchsatzraten der heterogenen Föderation, mit und ohne AM nicht signifikant voneinander.

5 Verwandte Arbeiten

Indem die Schnittstellenkonflikte zwischen ESP durch Einsatz von JEPC überwunden werden können, ist die Umsetzung einer Föderation heterogener ESP ein konsequenter Schritt. Ein Vorteil föderativer Systeme ist der Lastausgleich zwischen den beteiligten Systemen und der Leistungszugewinn durch eine Anfrageoptimierung. Wie im Abschnitt 4 gezeigt wurde, trifft dies ebenfalls für eine Föderation aus ESP zu. Eine gemeinsame Herausforderung von föderativen Informationssystemen ist die Integration heterogener Systeme. Ein wesentlicher Unterschied ist der Ansatz der Föderation. Währenddessen in föderativen Datenbanksystemen ein operatorbasierender Ansatz selten vorkommt [Con97], ist dies die erste Wahl für eine ereignisstromgetriebene Verarbeitung. Der operatorbasierende Ansatz ist eine Konsequenz aus der zugrundeliegenden theoretischen Basis von JEPC (vgl. [Krä07]) und dem Ereignisstromparadigma. Indem Operatoren einer Anfrage gemäß den Leistungsprofilen der ESP verteilt verarbeitet werden, wird die Verarbeitungsgeschwindigkeit für diese Anfrage insgesamt deutlich erhöht.

Der vorgestellte Föderationsdienst fokussiert sich auf die Integration von heterogenen ESP ohne hierbei auf vorhandene föderative Datenbanksysteme zurückzugreifen. Wie sich ESP in vorhandene Datenbankföderationen einbinden lassen, wird in [ea09] näher untersucht. Der Vorteil dieses Ansatzes ist klar: Ist einmal die Brücke zwischen der Datenstrom- und Datenbankwelt geschlagen, so profitiert ein föderatives Datenbank-/Datenstromssystem automatisch von den Ergebnissen jahrelanger Entwicklung und Forschung. Um konsistent mit dem Datenbankparadigma zu sein, werden kontinuierliche Anfragen als Ganzes an die beteiligten ESP weitergegeben. Hierdurch kann allerdings keine feingranulare Aufteilung der Anfrage unternommen werden und folglich wird nicht das gesamte Optimierungspotential ausgenutzt. Das föderative System dieser Arbeit setzt genau hier an. Indem Anfragen operatorweise verteilt werden, kann ein möglichst gute Anfrageoptimierung erreicht werden.

6 Zusammenfassung und zukünftige Arbeit

Für zeitkritische Anwendungen hat sich Ereignisstromverarbeitung längst etabliert. Fehlende Standards erschweren die Kompatibilität von CEP-Systemen untereinander. Dem kann durch Einsatz eines Adapters wie JEPC entgegengetreten werden. Dieser Ansatz ermöglicht ein operatorbasierendes föderatives CEP-System, das in dieser Arbeit konzeptionell besprochen und evaluiert wurde. Das Design dieses Systems erlaubt anstelle des standardmäßigen Entscheidungsbaums auch benutzerdefinierte Verfahren. Damit ist die

Verteilung adaptier- und austauschbar. Um die Kommunikation zwischen dem föderativen Systems und den beteiligten ESP zu verbessern, wird eine Autonomie-Maximierung durchgeführt, in deren Folge möglichst große Teilanfragen direkt durch den einzelnen ESP unabhängig bearbeitet werden können. Die Evaluation zeigt, dass heterogene Föderationen gegenüber homogenen Föderationen höhere Durchsatzraten erzielen. In Abhängigkeit der Anfragestruktur kann die Autonomie-Maximimierung zusätzliche Leistungsvorteile bringen.

Da der Fokus der aktuellen Arbeit auf einer prototypischen Entwicklung lag, berücksichtigt diese Architektur momentan keine Ausnahmebehandlungen im Fall eines Ausfalls beteiligter Komponenten. Dies kann jedoch in zukünftigen Versionen durch Techniken wie Checkpoints oder redundanter Datenhaltung, sowie einem Adhoc-Master nachgeholt werden.

Das vorgestellte Klassifikationsverfahren findet eine gute Startplazierung der Operatoren, aber berücksichtigt laufende Änderungen nicht. Zur Laufzeit können allerdings weitere Verbesserungen im Rahmen einer dynamischen Planmigration unternommen werden. Die hierzu nötigen Konzepte und Techniken im Rahmen einer Föderation sind weitere spannende Forschungsfragen.

Literatur

[BW01] Shivnath Babu und Jennifer Widom. Continuous Queries over Data Streams. *SIGMOD Rec.*, 30(3):109–120, September 2001.

[Con97] Stefan Conrad. *Föderierte Datenbanksysteme: Konzepte der Datenintegration*, Jgg. 1 of *3-540-63176-3*. Springer-Verlag, Berlin/Heidelberg, September 1997.

[ea09] Irina Botan et al. Design and Implementation of the MaxStream Federated Stream Processing Architecture. Technical Report TR-632, ETH Zurich Department of Computer Science, June 2009.

[HGMR13] Bastian Hoßbach, Nikolaus Glombiewski, Andreas Morgen und Bernhard Ritter, Franz und Seeger. JEPC: The Java Event Processing Connectivity. *Datenbank-Spektrum*, 13(3):167–178, 2013.

[JMS+08] Namit Jain, Shailendra Mishra, Anand Srinivasan, Johannes Gehrke, Jennifer Widom, Hari Balakrishnan, Uğur Çetintemel, Mitch Cherniack, Richard Tibbetts und Stan Zdonik. Towards a Streaming SQL Standard. *Proc. VLDB Endow.*, 1(2):1379–1390, August 2008.

[Krä07] Jürgen Krämer. *Continuous Queries over Data Streams - Semantics and Implementation*. Dissertation, Fachbereich Mathematik und Informatik, Philipps-Universität Marburg, 2007.

[Luc01] David C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. 0201727897. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[ZRH04] Yali Zhu, Elke A. Rundensteiner und George T. Heineman. Dynamic Plan Migration for Continuous Queries over Data Streams. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, Seiten 431–442, New York, NY, USA, 2004. ACM.

Unterstützung von datengetriebenen Prozessschritten in Simulationsstudien durch Verwendung multidimensionaler Datenmodelle

Monika Walter, Prof. Dr.-Ing. Axel Hahn

Business Engineering Department für Informatik
Universität Oldenburg
Ammerländer Heerstraße 114-118
26129 Oldenburg
walter.monika@gmx.de
hahn@offis.de

Abstract: Die vorliegende Arbeit beschäftigt sich mit Ansätzen zur Unterstützung von datengetriebenen Prozessschritten in Simulationsstudien der diskreten Event-Simulation. Der Lösungsansatz besteht darin, die in Data-Warehouse-Systemen vorhandenen Konzepte zur optimierten Nutzung großer Datenmengen aus operativen Systemen unter Anwendung multidimensionaler Datenmodelle für die Simulationsstudien zugänglich zu machen. Der Schwerpunkt der Arbeit liegt in der Erstellung eines Konzeptes, welches Aspekte zu Verbesserungspotenzialen für die drei Bereiche Datenbeschaffung, Initialisierung und Parametrisierung von Simulationsstudien sowie Reduzierung von wahrscheinlichkeitsbasiertem Dateninput betrachtet. Ziel des Ansatzes ist es, die bereits in den Data-Warehouse-Systemen umgesetzten Aggregations- und Abstraktionsmechanismen für Simulationsstudien systematisiert zugänglich zu machen und somit den Input großer Datenmengen für diese zu ermöglichen.

1 Einleitung

Durch die Verwendung von IT-Systemen auf operativer Ebene erzeugen Unternehmen eine Flut von Daten (vgl. [BG09], S. 6). Die konsequente Verwendung dieser Daten zur Analyse bzw. Optimierung der Prozesse und zur Unternehmensplanung als Basis in der Entscheidungsfindung auf strategischer, mittelfristiger und operativer Ebene rückt für die Unternehmen immer stärker in den Fokus (vgl. [Ga10], S. 47f.). Im Rahmen von Simulationsstudien können etwa Daten zur Abbildung der Systemlast aus diesen operativen Systemen, beispielsweise einem Logistiksystem oder weiteren vorhandenen operativen Datenquellen wie BDE- oder PPS-Systemen, übertragen und orientiert an der abzubildenden Fragestellung aufbereitet werden. Im Simulationssystem können Prozesse reproduzierbar definiert werden, daher stellen vor allem die Unternehmensbereiche der Produktion und Logistik einen Schwerpunkt für den Einsatz der Simulation zur Planungsunterstützung dar (vgl. [VDI10]).

Die Verwendung von Methoden und Werkzeugen der *Business Intelligence* (BI) zur Analyse der Unternehmensperformance hat sich als integrativer Gesamtansatz durchgesetzt. Betriebswirtschaftliche Analysewerkzeuge wie die Balanced Scorecard sind weitestgehend etabliert und in den strategischen Bereichen bedient sich das Unternehmenscontrolling entsprechender Kennzahlensysteme zur Bewertung der Key-Performance-Indikatoren (KPIs). Der Nutzen solcher Analysen – vor allem in den vertrieblichen und finanziellen Unternehmensbereichen – ist unbestritten und lässt sich durch verbesserte Unternehmensergebnisse, optimierte Prozesse und steigende Renditen messen. In diesen Analysewerkzeugen finden häufig die in den operativen IT-Systemen erzeugten Daten Verwendung.

Die drei aufgeführten betrieblichen Instrumente *(Simulationsstudien, Daten aus operativen IT-Systemen sowie Methoden und Werkzeuge der BI)* im Rahmen der betrieblichen Entscheidungsunterstützung miteinander zu integrieren, ist eine seit langem diskutierte Fragestellung (vgl. [Ze92], S. 1).

In einem Beispiel aus der Intralogistik soll die Diskrepanz zwischen Theorie und Praxis kurz erläutert werden: Ein mittelständisches Produktionsunternehmen plant den Einsatz eines fahrerlosen Transportsystems zur Optimierung des werksinternen Transports von Material als Ersatz für Staplertransporte. Zur Prüfung der Wirtschaftlichkeit wird eine Simulationsstudie in Auftrag gegeben. Aus dem ERP-System werden die Daten in Form von Einzeltransaktionen zu den bisher eingesetzten Staplertransporten für das zu überplanende Areal für die Zeitdauer von ca. 1 Jahr extrahiert und zur Verfügung gestellt. Die Aufbereitung der Daten für die Simulation stellt sich als so aufwendig heraus, dass letztlich für die Simulation nur Daten von ca. 5 Werktagen eingesetzt werden. Die Aussagekraft der Simulationsergebnisse wird vom Untenehmen entsprechend kritisch gesehen. Das Beispiel zeigt, dass Simulationsexperten häufig nicht in der Lage sind, die großen Mengen der zur Verfügung stehenden Daten aufzubereiten und zu verwenden.

2 Ausgangslage und Problemstellung

Der nachfolgend beschriebene Ansatz widmet sich der Fragestellung, ob die Verwendung von in Data-Warehouse-Systemen in multidimensionalen Datenmodellen abgelegten Daten für Simulationsstudien systematisch Vorteile bringt. Im Verlauf der Durchführung von Simulationsexperimenten soll eine Durchführungsoptimierung dieser Studien erreicht werden. Vorteile können sich insofern ergeben, da auf große Datenmengen zurückgegriffen werden kann, die bereits im Data Warehouse aggregiert wurden.

Eine erste Problematik in der Durchführung von Simulationsstudien besteht in der Datenbeschaffung. Im Prozess nimmt die Beschaffung und Aufbereitung von Daten einen Umfang von 30 % bis 50 % des Gesamtaufwands ein (vgl. [RSW08], S. 45; [KG95], S. 110). Dabei werden verschiedene Arten von Daten unterschieden: Daten zur Beschreibung der Topologie und Struktur des Systems, zur Beschreibung der Komponenten und der Zustandsänderungen sowie Daten für die Abbildung der Systemlast (vgl. [VDI10]). Aufgrund der Vielzahl der möglichen Datenquellen sind bei der Datenaufbereitung regelmäßig Probleme bezüglich der Datenkonsistenz sowie der Datengenauigkeit zu beo-

bachten. Die mangelnde Strukturierung der Inputdaten führt zu Problemen bei der Identifizierung eines passenden Verteilungstypen sowie bei der Schätzung der Parameter der Verteilung.

Die zweite Problematik ist in der Initialisierungsphase von Simulationsexperimenten zu lokalisieren. Zur Definition einer belastbaren Anzahl von Simulationsläufen ist es notwendig, die Inputdaten zu verifizieren und zu strukturieren. Unter dem Begriff der Initialisierungsphase wird der Zeitraum verstanden, den ein Simulationssystem benötigt, um den Zustand zu erreichen, in dem verwendbare Ergebnisdaten erzeugt werden. Dieser ist erreicht, wenn der Mittelwert der Beobachtungen dem Erwartungswert für unendlich viele Beobachtungen entspricht (vgl. [Kü06], S. 409). Dabei hängt die Genauigkeit eines Simulationsexperiments u. a. von der Anzahl der Simulationsläufe ab (vgl. [KS99], S. 3). Die Verkürzung der Initialisierungsphase basierend auf der Optimierung des Dateninputs ist also anzustreben.

Die dritte Problematik ergibt sich daraus, dass Simulationsstudien mit einer hohen Anzahl von Zufallsvariablen, die aufgrund fehlender bzw. mangelhaft strukturierter Inputdaten verwendet werden, die Ergebnisqualität verringern bzw. die Anzahl der Simulationsexperimentwiederholungen steigen lassen, um aussagefähige Ergebnisse zu erhalten. Gleiches gilt für wahrscheinlichkeitsbasierten Simulationsinput, der die Schwierigkeit zur Konkretisierung der ableitbaren Schlussfolgerungen erhöht.

Ausgehend vom operativen System wird nach Übertragung und Abstraktion von Daten in das Data-Warehouse-System das Ziel verfolgt, die Unternehmensführung in der Entscheidungsfindung durch die Darstellung logistischer KPIs zu unterstützen. Die Übertragung der Daten vom logistischen System in das Data Warehouse wird unter dem Begriff des ETL-Prozesses (Extraktion, Transformation, Laden) vorgenommen (vgl. [BG09], S. 38). Integriert in diesen ETL-Prozess ist eine Aggregation der Daten. Die im operativen System, häufig in relationalen Datenbanken, abgebildete Speicherung der Daten in normalisierter Form wird innerhalb des ETL-Prozesses aufgegeben, stattdessen erfolgt im Data Warehouse die Speicherung in multidimensionaler Form, denormalisiert und nach definierten Regeln aggregiert in Form von Dimensionen und Fakten. Von multidimensionalen Datenstrukturen wird gesprochen, wenn mehrere Dimensionen die Charakteristika eines Faktums beschreiben. In der Intralogistik sind typische Dimensionen die Zeit (z.B. Zeitpunkt eines Pickvorgangs, eines Transports oder einer Lieferung) und der Ort (z.B. Lagerort). Als relevante Fakten im Prozess werden häufig Mengen, Chargen, Ladehilfsmittel oder Aufträge betrachtet (vgl. [Wa11]). Innerhalb des multidimensionalen Datenmodells sind die häufigsten Speicherformen das Star Schema sowie das Snowflake Schema (vgl. [BG09], S. 214ff.).

Die bei der Übertragung der Daten vom operativen System in das Data Warehouse bzw. in das Simulationssystem verwendeten Abstraktionsregeln und Aggregationsmechanismen beinhalten vergleichbare Aufgabenstellungen. Der Begriff der Abstraktion beschreibt Eigenschaften auf der Strukturebene. Die Unterteilung umfasst die Makro-, Meso- und Mikroebene (vgl. [KMU06], S. 147 ff.). Kemper et al. verwenden die Makro- und Mikroebene für die Definition eines Vorgehensmodells innerhalb der BI, die strukturellen Ebenen können, ergänzt um eine Mesoebene, auf die Abstraktion übertragen

werden. Auf der Mikroebene werden einzelne Objekte der zu analysierenden Systeme und deren direkte Beziehungen zueinander betrachtet (z.B. Maschinen, Transporteinheiten, detaillierter Warenfluss). Auf der Mesoebene werden organisatorische Einheiten und deren Beziehungen betrachtet (z. B. Lagerstrukturen, Transportsysteme), auf der Makroebene können komplexere Subsysteme inklusive ihrer Interaktionen eingeordnet werden (z. B. Werksplanung, Warenflussplanung). Die Darstellung bestimmter zeitlicher Einheiten ist unter dem Begriff der Aggregation abgebildet. Die betriebswirtschaftliche Einteilung in kurz-, mittel- und langfristige Betrachtungszeiträume wird in ein Verhältnis zum Abstraktionsgrad gesetzt. Die Grenzen sowohl zwischen den Abstraktions- als auch den Aggregationsebenen sind fließend und sollen auch nicht starr abgegrenzt werden.

Der Zusammenhang zwischen den Abstraktions- und Aggregationsebenen wird bezogen auf das in Kapitel 1 eingeführte Beispiel deutlich, da die Erwartungshaltung des Unternehmens gewesen wäre, die Daten für das Werk bzw. das betroffene Areal auf Monats- bzw. Jahresebene zu betrachten (hohe Abstraktion, hohe Aggregation). Die im Beispiel aufgeführte Zeitspanne weniger Tage stellt dagegen eine niedrige Aggregation dar.

3 Zielsetzung und Vorgehensweise

Ziel der Arbeit ist die Erstellung eines Konzeptes mit drei Komponenten. Das Konzept dient der Verknüpfung der Ansätze des Data Warehousing und der Simulation. Es besteht aus einem Ordnungskatalog, einem Fragenkatalog sowie einer Verwendungsanweisung für die Nutzung der beiden erstgenannten sowie die Bewertung der Ergebnisse. Die drei Komponenten sollen die Entscheidungsfindung unterstützen, ob in konkreten Simulationsfragestellungen die Optimierung der drei beschriebenen Problematiken möglich ist und Handlungsempfehlungen zur Umsetzung beinhalten sowie Technologien aus den betrachteten Themenbereichen berücksichtigen. Das Konzept richtet sich somit beispielsweise an Simulationsexperten und soll diesen die beschriebene Entscheidungsfindung ermöglichen.

Die Vorgehensweise zur Erstellung des Konzeptes teilt sich in die Identifikation von Elementen sowie in den Aufbau des Ordnungs- und Fragenkataloges und der korrespondierenden Verwendungsanweisung auf. Für den Aufbau des Ordnungskataloges erfolgte die Recherche nach relevanten Elementen in Form einer Literaturanalyse sowie durch Auswertung bestehender Konzepte innerhalb der Business Intelligence und der Simulation. Die Elemente werden durch Eigenschaften und Attribute beschrieben. Dabei werden sowohl die zeitliche Ebene in Form von Aggregationsprozessen als auch die strukturelle Ebene der involvierten Themenbereiche in Form von Abstraktionsprozessen integriert.

Die Vorgehensweise zur Umsetzung gliedert sich in mehrere Phasen. In *Phase 1* sind innerhalb der dargestellten Themenbereiche (operativ, analytisch, simulativ) relevante Elemente mit ihren Eigenschaften und Attributen unabhängig voneinander zu identifizieren, klassifizieren und zu strukturieren. *Phase 2* beinhaltet die Erstellung der drei Komponenten sowie die Beschreibung der enthaltenen Elemente anhand von Beispielen bezogen auf die Inhaltsdomäne Intralogistik. Das entstandene Konzept wird in der *Phase 3*

validiert, um die Verwendung in Simulationsprojekten zu prüfen. Dazu gehört ebenfalls die Identifikation zeitlicher und wirtschaftlicher Vorteile durch die Verwendung im Simulationsumfeld.

Die beschriebene Zielsetzung bzw. die erwarteten Ergebnisse sind gegenüber bestehenden Ansätzen abzugrenzen. Bei der Analyse vorliegender Veröffentlichungen zum Themenkreis Simulation sind Forschungsvorhaben, welche die werkzeuggestützte Optimierung des Simulationsprozesses und die Teil- bzw. Automatisierung des Simulationsprozesses zum Gegenstand haben, als zwei Schwerpunktthemen zu identifizieren. Die Forschungsprojekte AssistSim und EDASim befassen sich mit der Erarbeitung von Assistenzfunktionen für Simulationsstudien (vgl. [MSW12]). Ein Forschungsprojekt der Universität Ilmenau beschäftigt sich mit der automatischen Generierung von Simulationsmodellen für Produktionssysteme (vgl. [Be11]). In Abgrenzung zu diesen Projekten fokussiert die im vorliegenden Artikel beschriebene Forschungsarbeit auf die von außen auf die Simulation gerichtete Betrachtung der Prozesse bzw. Systeme und fokussiert innerhalb der Simulation auf qualitativ bewertbare und systematisierbare Fragestellungen bezogen auf die datengetriebenen Aspekte von Simulationsstudien. Im Themenkreis der BI finden sich Fragestellungen zur Nutzung von Daten aus Data-Warehouse-Systemen für den Aufbau von Frühwarn- und Früherkennungssystemen. Die Nutzung dieser Systeme kann sich für Unternehmen als Wettbewerbsvorteil entwickeln, die Ausführungen betrachten jedoch nicht die Verwendung dieser Daten explizit in Simulationsstudien (vgl. [Ge05]).

4 Aktueller Stand der Untersuchungen

In diesem Kapitel wird der aktuelle Stand bei der Entwicklung der beiden Kataloge inkl. eines Beispiels dargestellt. Dabei steht die Vorstellung der Struktur des Ordnungskatalogs im Vordergrund. Diese Struktur muss die verschiedenen Aspekte der beteiligten Themenbereiche wie Gestaltung, Steuerung, Funktionalität, Transformation sowie die jeweils gewählte Inhaltsdomäne berücksichtigen. Sowohl die beiden Kataloge als auch die Verwendungsanweisung dürfen nicht als abschließend zu definierende Komponenten verstanden werden. Basierend auf der Betrachtung weiterer Inhaltsdomänen, spezieller Fragestellungen, Anforderungen oder Systemumgebungen soll die Möglichkeit bestehen, variable Ergänzungen vorzunehmen. Daher sind auch Aspekte der Erweiterbarkeit für das Konzept zu berücksichtigen bzw. zu betrachten. Insbesondere sind die Abstraktions- und Aggregationsebenen für verschiedene Anwendungsszenarien zu prüfen und bei Bedarf anzupassen. Diese Möglichkeit zur Modifikation ist im Konzept vorgesehen, solange die beschriebenen Grundstrukturen eingehalten werden.

4.1 Struktur des Ordnungskatalogs

Der Ordnungskatalog gliedert sich in zwei Ebenen. In jeder Ebene wird durch eindeutige Kennzeichnung in Form von Ordnungszahlen, Eigenschaftszahlen und Ausprägungszahlen die Identifikation der Elemente, Eigenschaften und Ausprägungen ermöglicht.

Ebene 1 stellt fünf Perspektiven dar. Innerhalb dieser Perspektiven werden auf Ebene 2 Aspekte unterschieden. Innerhalb des Klassifikationssystems (Perspektive und Aspekte) werden die Elemente sequentiell durchnummeriert.

Ebene 1: Perspektive

Die Aufteilung des Ordnungskatalogs auf oberster Ebene in Perspektiven orientiert sich daran, dass die identifizierten Elemente aus den Themenbereichen Data Warehousing, Simulation und Intralogistik anhand jeweils eines wesentlichen Aspektes einer Perspektive zugeordnet werden können.

Die **Gestaltungsperspektive** beinhaltet Elemente, welche die zu analysierende Fragestellung (z. B. einer Simulationsstudie oder Modellierung) gestalten. Die Elemente bilden die Grundstrukturen der Fragestellung der konkreten Simulationsstudie ab. In der Gestaltungsperspektive sind auch Elemente zugeordnet, die zur Klärung von Modellierungsaspekten sowie zur Phaseneinordnung der Simulationsstudie beitragen. Die **Steuerungsperspektive** enthält die Simulation steuernde Elemente, die Entscheidungskriterien wie beispielsweise die Art des Zeitfortschrittsmechanismus, den zu betrachtenden Zeithorizont, die Simulationsmethode oder Konfigurationsregeln zum Gegenstand haben. Die **Basisperspektive** enthält neben weiteren Konfigurationselementen auch statische Elemente ohne Konfigurationseigenschaft, welche aus Informationsgründen im Katalog enthalten sind. Dabei kann es sich um konkrete Elemente wie z. B. um Phasenschritte im Vorgehensmodell und funktionale Elemente handeln. Im Ordnungskatalog sind diese zum Einen enthalten, um den Empfehlungscharakter der Ergebnismenge mit möglichst konkreten Informationen anzureichern. Zum Anderen dienen sie als Basis für die inhaltlich konkretisierten Elemente der Domänenperspektive. Die **Transformationsperspektive** dient zur Überführung des multidimensionalen Datenmodells in die Strukturen des Dateninputs für Simulationen. Sie beinhaltet Aspekte zu Dimensionen und Fakten und überträgt deren Konzepte in Subjekte, Objekte und Instrumente eines Simulationsmodells.

Die **Domänenperspektive** besteht aus relevanten Aspekten der gewählten Inhaltsdomäne Intralogistik. Diese Perspektive kann zukünftig um weitere Inhaltsdomänen erweitert werden. In dieser Perspektive sind beispielsweise auch Datenselektionselemente eingeordnet, die dazu führen, dass z. B. eine Eingrenzung von Daten aus dem Data Warehouse oder dem operativen System erfolgt, bezogen auf die exemplarisch definierten Dimensionen <ZEIT> (Aggregationsprozesse) und <ORT> (Abstraktionsprozesse).

Ebene 2: Aspekte der jeweiligen Perspektive

Die Ebene 2 unterteilt sich in die als führend bzw. ausschlaggebend für die Einordnung der Elemente definierten Aspekte. Zu den beschriebenen Perspektiven werden jeweils disjunkte Aspekte zugeordnet, welche die relevanten Kriterien darstellen. Aspekte der Gestaltungsperspektive sind z. B. Modellaspekte, Aspekte der Steuerungsperspektive sind administrative, zeitliche, infrastrukturelle, technische sowie äußere Einflüsse. Innerhalb der Basisperspektive werden Aspekte z. B. strukturiert in Aktivitäten, Objekte und Funktionen. Die Domänenperspektive spiegelt die jeweils für die Fragestellung

relevanten Aspekte wider. In der initialen Befüllung enthält sie Elemente der Intralogistik.

Eigenschaften und Ausprägungen

Unterhalb der Ebenen 1 und 2 sind für das zu beschreibende Element Eigenschaften und deren Ausprägungen definiert. Wesentliche Eigenschaften der Elemente sind die Abstraktion und die Aggregation. Neben den strukturellen und den zeitlichen Eigenschaften können für Elemente auch andere Eigenschaften und Ausprägungen zutreffen. Vor allem bei Elementen der Gestaltungsperspektive wird die Eigenschaft der *Variante* verwendet um Beispiele aufzunehmen. Innerhalb jeder Eigenschaft sind die Ausprägungen disjunkt zueinander zu verstehen.

Die gesamte Struktur des Ordnungskatalogs ist in folgender Abbildung zusammengefasst:

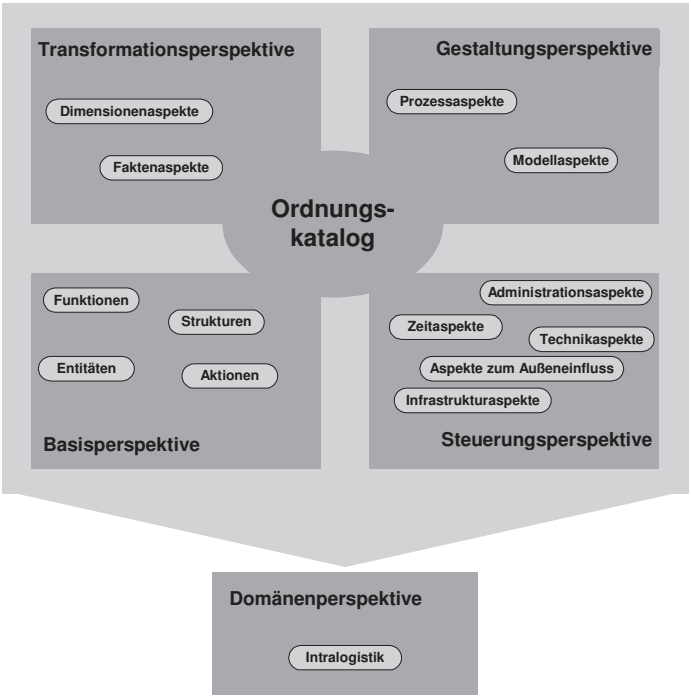


Abbildung 1: Aufbau des Ordnungskatalogs

4.2 Beispielhafte Darstellung eines Elements

Das Element *1.1-2 Modellart* führt zu der Entscheidung, um welche Art von Simulationsmodell es sich handelt. Im Ordnungskatalog sind jedem Element erklärende Erläuterungen vorangestellt. Auf diese Darstellung dieser Erläuterungen wird in diesem Bei-

spiel verzichtet, da sie zum Verständnis des Zusammenhangs zwischen Element und Frage nicht relevant sind.

1.1	2	Modellart	Arten von Simulationsmodellen	
E1		Varianten		--
		A	Statisches Modell	(1.1-2-E1-A) xor (1.1-2-E1-B:F)
		B	Zeitorientiertes diskretes Simulationsmodell (deterministisch)	(1.1-2-E1-B) xor (1.1-2-E1-A,C:F)
		C	Ereignisorientiertes diskretes Simulationsmodell (deterministisch)	(1.1-2-E1-C) xor (1.1-2-E1-A:B,D:F)
		D	Zeitorientiertes diskretes Simulationsmodell (stochastisch)	(1.1-2-E1-D) xor (1.1-2-E1-A:C,E:F)
		E	Ereignisorientiertes diskretes Simulationsmodell (stochastisch)	(1.1-2-E1-E) xor (1.1-2-E1-A:D, F)
		F	Keine der Varianten	(1.1-2-E1-F) xor (1.1-2-E1-A:E)

Abbildung 2: Beispielelement Ordnungskatalog Modellart

Das dazugehörende Element im Fragenkatalog dient der Auswahl eines der Entscheidungskriterien:

1.2	Welche Art von Simulationsmodell soll entwickelt werden?		
	Frage muss beantwortet werden, es gibt nur eine Antwortmöglichkeit.		
Antwortmöglichkeiten	A	Statisch	(1.1-2-E1-A)
	B	Zeitorientiert, diskret, deterministisch	(1.1-2-E1-B)
	C	Ereignisorientiert, diskret, deterministisch	(1.1-2-E1-C)
	D	Zeitorientiert, diskret, stochastisch	(1.1-2-E1-D)
	E	Ereignisorientiert, diskret, stochastisch	(1.1-2-E1-E)
	F	Keine der Varianten	(1.1-2-E1-F)

Abbildung 3: Beispielelement Fragenkatalog Modellart

Mit der Beantwortung der Frage im Fragenkatalog ist sowohl die Auswahl eines Filterkriteriums für das (jeweilige) Element verbunden als auch eine Gewichtung der Antwort. Nach Beantwortung aller Fragen im Fragenkatalog kann damit sowohl eine Filterung der relevanten Elemente vorgenommen werden als auch eine Gewichtung dieser Elemente mit Hinblick auf eine Bewertung, ob die Verwendung multidimensionaler Datenmodelle im konkreten Fall einer Simulationsstudie eine Optimierung bedeuten kann. Im Beispiel wird die Antwortmöglichkeit E gewählt. Dies führt für die Auswahl einer Modellart zu

einer Verdeutlichung der Aufgabenstellung. Ein Ausschlusskriterium zur Verwendung des beschriebenen Ansatzes ergibt sich, wenn keine der aufgeführten Modellarten relevant ist, da das Konzept nur für diese betrachtet wurde. Der Ausschnitt aus der Verwendungsanweisung für die Bewertung des Elements bezogen auf das Beispiel aus Kapitel 1 (ereignisorientiertes diskretes Simulationsmodell) ist in Abbildung 4 dargestellt.

Element	Name Eigenschaft	Ausprägung	Frage	Bewertung	Bewertungshinweis Bietet die Verwendung multidimensionaler Datenmodelle Unterstützungspotenzial für die datengetriebenen Prozessschritte der Simulationsstudie?
1.1-2	Modellart E1 Varianten	A Statisches Modell	1.2	10	Eine Simulationsstudie wird i. d. R. nicht als statisches Modell umgesetzt. Wenn diese Antwort ausgewählt wurde, ist sowohl das Instrument der Simulationsstudie als auch die Verwendung multidimensionaler Datenmodelle als Input zu überdenken. Die Verwendung der beiden Komponenten kann nur bedingt empfohlen werden.
1.1-2	Modellart E1 Varianten	B Zeitorientiertes diskretes Simulationsmodell (deterministisch)	1.2	30	Eine Simulationsstudie, die als zeitorientiertes diskretes Simulationsmodell mit deterministischen Eingabedaten realisiert wird, wird durch die Verwendung multidimensionaler Datenmodelle unterstützt.
1.1-2	Modellart E1 Varianten	C Ereignisorientiertes diskretes Simulationsmodell (deterministisch)	1.2	30	Eine Simulationsstudie, die als ereignisorientiertes diskretes Simulationsmodell mit deterministischen Eingabedaten realisiert wird, wird durch die Verwendung multidimensionaler Datenmodelle unterstützt.
1.1-2	Modellart E1 Varianten	D Zeitorientiertes diskretes Simulationsmodell (stochastisch)	1.2	50	Eine Simulationsstudie, die als zeitorientiertes diskretes Simulationsmodell mit (teilweise) stochastischen Eingabedaten realisiert wird, wird durch die Verwendung multidimensionaler Datenmodelle in besonderem Maße unterstützt, da die Reduzierung der stochastischen Elemente eine Aufwandsreduzierung bedeutet.
1.1-2	Modellart E1 Varianten	E Ereignisorientiertes diskretes Simulationsmodell (stochastisch)	1.2	50	Eine Simulationsstudie, die als ereignisorientiertes diskretes Simulationsmodell mit (teilweise) stochastischen Eingabedaten realisiert wird, wird durch die Verwendung multidimensionaler Datenmodelle in besonderem Maße unterstützt, da die Reduzierung der stochastischen Elemente eine Aufwandsreduzierung bedeutet.
1.1-2	Modellart E1 Varianten	F Keine der Varianten	1.2	0	Das Konzept wurde nur für die in den Modellarten der in A-E dargestellten Varianten betrachtet. Für weitere Modellarten wurde die Verwendung multidimensionaler Datenmodelle nicht betrachtet, daher kann die Verwendung nicht empfohlen werden.

Abbildung 4: Bewertung des Elements Modellart

Insgesamt bestehen der Ordnungskatalog zum derzeitigen Stand aus 38 Elementen und der Fragenkatalog aus 19 Fragen. Bei über der Hälfte der Elemente sind die Eigenschaften der Abstraktion und der Aggregation zugeordnet. Ergibt sich aus der Beantwortung der Fragen im Fragenkatalog für den Simulationsexperten eine Eignung von Daten aus multidimensionalen Datenmodellen für die konkrete Simulationsfragestellung, ist dieser aufgrund des Empfehlungscharakters der Ergebnisse in der Lage, die Daten auf dem ausgewählten Niveau der Abstraktion (strukturell) und Aggregation (zeitlich) aus einem Data Warehouse zu extrahieren oder entsprechend der Regeln multidimensionaler Datenmodelle aufzubereiten.

5 Fazit

In diesem Beitrag wird ein Ansatz beschrieben, die datengetriebenen Prozessschritte von Simulationsstudien durch Verwendung von großen Datenmengen aus multidimensionalen Datenmodellen (Data-Warehouse-Systemen) zu verbessern. Die Qualität der Aussagen, die aus Simulationsstudien gewonnen werden können, hängt in hohem Maße von der Qualität der verwendeten Daten bzw. dem Dateninput ab. Die Verwendung der mul-

tidimensionalen Daten bietet aus Sicht der Autorin ein hohes Potenzial bezogen die Aspekte der Verbesserung der Datenqxualität z. B. durch Verringerung der Verwendung von Wahrscheinlichkeitswerten, eine Reduzierung des Erhebungsaufwands, da auf aufbereitete Daten systematisiert zugegriffen sowie die Strukturiertheit der verwendeten Daten in den Simulationsstudien effektiv genutzt werden kann.

Der Ansatz soll dazu beitragen, die Szenarien zur Verwendung von Data-Warehouse-Anwendungen um die Nutzung in Simulationsstudien zu erweitern. Vor allem Unternehmen kleiner und mittlerer Größe haben die beiden Themenbereiche noch nicht hinreichend für sich erschlossen. Der vorgestellte Ansatz dient dazu, die Integration zu vereinfachen und den Simulationsexperten in die Lage versetzen, die Aufbereitung und Nutzung der Daten basierend auf der konkreten Fragestellung in Zusammenarbeit mit Data Warehousing-Experten zu optimieren. In der Praxis gilt es, dieses Potenzial zu konkretisieren und die Komponenten des Konzeptes zu erproben. Dazu sollten zur Validierung des Konzeptes die beiden Kataloge auf beispielhafte Fragestellungen und Szenarien von intralogistischen Simulationsstudien angewandt werden, um die Optimierungspotenziale durch die Anwendung des Konzeptes zu überprüfen.

Literaturverzeichnis

[Be11] Bergmann, S.: Automatische Generierung adaptiver und lernfähiger Modelle zur Simulation von Produktionssystemen. TU Ilmenau, publiziert: Doctoral Consortium, der 10. Internationalen Tagung Wirtschaftsinformatik, Zürich, 2011.

[BG09] Bauer, A.; Günzel, H.: Data Warehouse Systeme. Dpunkt, Heidelberg, 2009.

[Ga10] Gansor et. al: Gansor, T.; Totok, A.; Stock, S.; Baars, H.: Von der Strategie zum Business Intelligence Competency Center (BICC). Hanser, München, 2010.

[Ge05] Gehra, B.: Früherkennung mit Business-Intelligence-Technologien, Anwendung und Wirtschaftlichkeit der Nutzung operativer Datenbestände. Deutscher Universitäts-Verlag, Wiesbaden, 2005.

[KMU06] Kemper, HG; Mehanna, W.; Unger, C.: Business Intelligence. Vieweg, Wiesbaden, 2006.

[KG95] Košturiak, J.; Gregor, M. (1995): Simulation von Produktionssystemen. Springer, Berlin.

[KS99] Küll, R.; Stähly, P.: Zur Planung und effizienten Abwicklung von Simulationsexperimenten. In: (Biethahn, Jörg; Hummeltenberg, Wilhelm; Schmidt, Bernd; Stähly, Paul; Witte, Thomas, Hrsg.): *Simulation als betriebliche Entscheidungshilfe*. Physica Verlag, Heidelberg, 1999, S. 1-21.

[Kü06] Kühn, W.: Digitale Fabrik. Fabriksimulation für Produktionsplaner. Hanser, München, 2006.

[MSW12] Mayer, G.; Spieckermann, S.; Wenzel, S.: Steigerung der Produktivität in Simulationsstudien mit Assistenzwerkzeugen. ZWF, Jahrgang 107, 2012.

[RSW08] Rabe, M.; Spiekermann, S.; Wenzel, S.: Verifikation und Validierung für die Simulation in Produktion und Logistik. Springer, Berlin, 2008.

[VDI10] VDI: Richtlinie 3633: Simulation von Logistik-, Materialfluss- und Produktionssystemen. Grundlagen. Blatt 1 Entwurf. Beuth, 2010.

[Wa11] Walter, M.: Standardisierte Kennzahlensysteme in der Intralogistik – Nutzen und Optimierungspotenzial erkennen. In: Praxishandbuch Logistik, Köln, 2011.

[Ze92] Zell, M.: Simulationsgestützte Fertigungssteuerung. Oldenbourg, München, 1992.

DduP – Towards a Deduplication Framework utilising Apache Spark

Niklas Wilcke

Datenbanken und Informationssysteme (ISYS)
University of Hamburg
Vogt-Koelln-Strasse 30
22527 Hamburg
1wilcke@informatik.uni-hamburg.de

Abstract: This paper is about a new framework called DeduPlication (DduP). DduP aims to solve large scale deduplication problems on arbitrary data tuples. DduP tries to bridge the gap between big data, high performance and duplicate detection. At the moment a first prototype exists but the overall project status is work in progress. DduP utilises the promising successor of Apache Hadoop MapReduce [Had14], the Apache Spark Framework [ZCF+10] and its modules MLlib [Ml14] and GraphX [XCD+14]. The three main goals of this project are creating a prototype of the mentioned framework DduP, analysing the deduplication process about scalability and performance and evaluate the behaviour of different small cluster configurations.

Tags: Duplicate Detection, Deduplication, Record Linkage, Machine Learning, Big Data, Apache Spark, MLlib, Scala, Hadoop, In-Memory

1 Introduction

Duplicate detection is the problem to find all duplicates within a set of elements. Duplicate detection is a performance critical task because of its quadratic complexity. Therefore it seems to be a promising approach to scale horizontally by solving deduplication problems on a distributed environment. On the other hand there are some drawbacks when switching to a classic map reduce framework like Apache MapReduce. We want to avoid these drawbacks by utilising the Apache Spark framework.

The main contributions of this paper are as follows: First, we describe the problems inherent to deduplication frameworks based on Apache MapReduce. Second, we describe the Apache Spark framework and its benefits for deduplication, in particular. Third, we describe architectural concepts of our deduplication framework and detail how it will make use of existing libraries.

2 Related Work

The deduplication process described by Peter Christen in the book "Data Matching" [Chr12b] is the basis for this work. It consists of five steps which shall be briefly explained. The first step is the preprocessing bringing the data into a standardised form. To reduce the quadratic complexity it is followed by a search space reduction, which selects candidate pairs from the search space in an efficient way. Search space reduction is often done by blocking [Chr12a]. The members of each selected pair are compared by a previously chosen set of measures (one per feature) to determine their similarity. Having assigned such a similarity vector to each pair a classifier splits the set into duplicate pairs and no-duplicate pairs. Typically the classification step is followed by a clustering phase [Chr12b]. The last step is the evaluation of the result.

Kolb, Thor and Ram at the University of Leipzig published "Dedoop: Efficient Deduplication with Hadoop" [KTR12]. Dedoop is a Hadoop MapReduce based application to perform deduplication. It is not open source and still a prototype therefore it's hard to investigate or contribute to. Using Hadoop MapReduce as a platform for duplicate detection has the following drawbacks. Hadoop MapReduce is not designed for an interactive usage. It is made for big data batch processing. Especially in the elaborative usage of such a framework an interactive quick responding shell is an advantage. A shell would ease the way exploring the framework and developing own applications.

Optimisation is a huge topic in duplicate detection. For each step in the process described by Christen there are several different algorithms and each one can be parameterised with multiple parameters. Finding the right configuration for a particular data source is a complex problem. Hadoop MapReduce lacks the ability to cache intermediate results to use them for multiple alternatives.

Implementing complex algorithms with Hadoop MapReduce is also difficult because of its low level API. Having high level primitives for distributed computing would simplify development. Nevertheless Hadoop MapReduce is a mature standard for distributed batch processing.

Apache Pig is a high level API for Apache Hadoop. Scripts are written in a procedural language called Pig Latin. Apache Pig enables the user also to store intermediate results in-memory. There is also Apache Hive, which is a data warehousing API on top of Apache Hadoop. Apache Hive provides data access in a declarative way via HiveQL an SQL like querying language. Both high level APIs are not designed to integrate in an existing programming language. Therefor they are not optimised to build huge projects, because the code written in Pig Latin or HiveQL is very hard to refactor.

Apache Spark introduces the new concept of a Resilient Distributed Dataset (RDD). It creates a high level abstraction for distributed fault tolerant in-memory computation. One huge advantage of RDDs is to be storable in memory for later reuse. Especially iterative algorithms benefit from this fact but also a process pipeline gets accelerated. For that reason Spark doesn't depend on slow hard disk I/O assuming a proper cluster configuration. RDDs are mostly distribution transparent. As a result the RDD API feels very much like programming in a non distributed environment. There is no need to learn a new program-

ming language because Spark offers bindings for Scala[1], Java and Python. Apache Spark claims to be the first interactive map reduce like platform. Coming with an interactive shell and a local mode Spark enables the user to start experimenting very quick. Testing and executing Apache Spark applications during development is also very easy. There is no need for a running Spark cluster. Only the spark libraries needs to be present in the project class path and a local cluster will be started on application startup. Other big projects like Apache Mahout [Mah14] are moving from Hadoop MapReduce to Spark. That will enrich the set of available algorithms for machine learning. The Mahout project is also about to build a R[2] like Scala Domain Specific Language (DSL) for Linear Algebra.

Apache Storm and Apache Flink are also two map reduce like computing frameworks in the Apache universe. Apache Flink is a very young project that combines a map reduce like API for distributed computations with query optimisation known from parallel databases. It has also an interactive shell but is still an incubator project and therefore not that mature. Apache Storm is a distributed real time data processing system. Both of them do not offer that much library support like Spark does.

Apache Spark seems to be the most promising candidate for implementing DduP on top. With its lean high level functional API, in-memory approach and programming language integration it meets our demands. Further more it ships with an interactive shell and is very easy to get started with. It is under active development and cutting edge approach in map reduce like distributed computing. There is an ecosystem of several libraries growing up and other projects like Mahout are migrating to Spark.

3 Duplicate Detection Process utilising Apache Spark

We interpret duplicate detection in the most general way. The problem is to find all duplicates within a Set C of tuples called corpus. A tuple is a datum with n features separated by a given scheme. It's mathematical representation is a n-dimensional vector. Each dimension corresponds to a feature. One possible four-dimensional tuple scheme and an example instance could be the following one describing persons.

Person Scheme and Example Instance

```
(forname, lastname, birthdate, place of birth)

(Max, Mustermann, 01.01.1970, Musterstadt)
```

Our duplicate detection process only slightly differs from Christen's previously mentioned one. All tuples get parsed from its input sources and afterwards they get preprocessed.

[1] Scala is a multi paradigm programming language combining object orientation and functional aspects which is executable on a Java Virtual Machine.

[2] R is a programming language focusing on statistics

There are different generic preprocessing steps available. For instance there are trim, to lower and regular expression removal steps. The only allowed type for features is String at the moment. Also numeric features are represented by strings. Strings were chosen because common similarity measures are defined to operate on them. To find typos and other errors introduced by human interaction a string representation seems to be more sensible. Nevertheless other types would also be possible to integrate. Result of the preprocessing is a set of all tuples called corpus.

To reduce the quadratic search space a blocking step filters out possible duplicate pairs in subquadratic time. There are two algorithms implemented. Suffix Array Blocking is implemented by using the Spark core API. Sorted Neighbourhood Blocking is implemented using the SlidingRDD from MLlib. Implementing other sliding window algorithms is very easy utilising SlidingRDD. Blocking is the crucial step in the whole process. Achieving a reduction ratio which is too low results in a long runtime and high memory consumption. A higher reduction ratio corresponds in general to a lower recall and therefore a loss of duplicate pairs. This trade-off is an optimisation problem which at the moment needs to be solved manually. Result of the blocking are tuple candidate pairs.

Every resulting candidate pair is compared featurewise by a similarity measure. There are several different measures available and most of them are normalised to the interval $[0, 1]$. 0 stands for not equal and 1 represents equality. Whether a normalisation is needed depends on the classification algorithm. For every feature dimension a measure needs to be defined. Therefore it's possible to use different measures for different dimensions. All the different measures are imported from the Scala library stringmetric [Str14]. Result of this comparison called similarity step is a similarity vector attached to the each candidate pair.

The binary classification in duplicate and no-duplicate pairs is done by a previously trained decision model. MLlib provides a Decision Tree, a Naive Bayse classifier and a Support Vector Machine (SVM) with a linear kernel for decision modeling. Result of the classification are the two sets duplicates and no-duplicates. The duplicate set containing all observed duplicate pairs is converted into a graph using the Spark library GraphX. Two connected nodes in the graph are representing a duplicate tuple pair. To resolve inconsistencies regarding transitivity a clustering is applied on the graph. The clustering tries to resolve constellations like A is a duplicate of B and B is a duplicate of C but A is not a duplicate of C. The whole process and its intermediate data types are depicted in Figure 1.

In this section we want to discuss briefly how the newly implemented steps scale in a distributed environment. The parsing can be done completely in parallel. Every line respectively tuple representation can be parsed independently. This is also valid for the preprocessing step.

Blocking in general always needs to shuffle the data. It depends on the algorithm which kind of shuffle is needed. Sorted Neighbourhood [Chr12b] for instance needs to sort the whole list of tuples by a key derived from each tuple. As shown in "Minimal MapReduce Algorithms" [TLX13] this can be done by TeraSort in $O(n \cdot log(n))$. Having n nodes in the cluster the fraction of $\frac{n-1}{n}$ of the input tuples needs to be shuffled to different nodes in the average. $\frac{n-1}{n} \approx n$ is valid for large n. That means the whole input set has to be

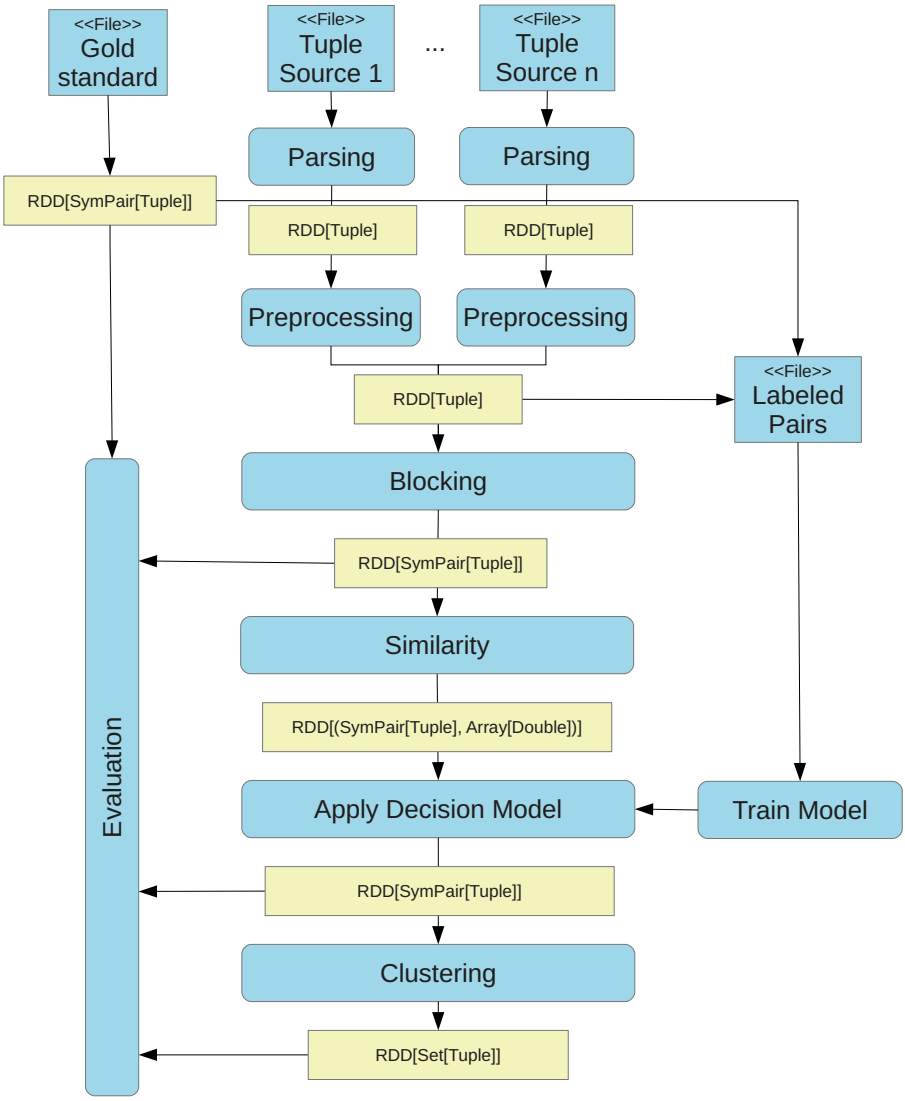


Figure 1: Overview of the separated steps (blue) of the duplicate detection process. The connecting input and output type is illustrated in yellow. RDD is the spark Resilient Distributed Dataset which can be seen as a distributed generic collection. SymPair is a generic symmetric pair.

relocated. Therefore the network bandwidth becomes an important factor.

Calculating the similarity of a tuple pair can be done in parallel. The classification task is completely covered by the existing Spark machine learning library MLlib. Clustering will be done via Sparks GraphX library. In the first step a graph is created with tuples as vertices and tuple pairs as edges. After deleting weak edges the transitive closure will be computed by GraphX. Every resulting component is a cluster of duplicates. Using a graph library for clustering might be an overhead, but the gain is a convenient API to implement cluster algorithms.

4 DduP Framework - Implementation Guideline

DduP aims to be a modular and open source distributed duplicate detection framework. It is implemented in Scala 2.10 utilising the Apache Spark framework.

Main guideline is minimalism and the usage of common standards, best practices and existing libraries. The project tries to focus on topics not covered yet. These topics are preprocessing, search space reduction, pipelining and evaluation in the context of distributed computing.

Figure 1 depicts the deduplication process which is a pipeline consisting of several steps called pipes. A pipe is a generic type combining an input and output type. A pipeline is a sequence of pipes with corresponding input and output types of the neighbouring steps. Steps of the same type can easily be replaced. Using this pattern it is easy to build modular steps for the different phases of the process. The resulting pipeline can be modified easily. This is especially useful for learning environments. There is a Java Framework "Tinkerpop Pipes" [Pip14] for building directed process pipelines. Due to its dependence on iterable collections it is not suitable to process RDDs. The last Spark 1.2 release included an alpha version of a pipelining framework for MLlib. Drawbacks of that framework are, that only SchemaRDDs are allowed to be used what may be inefficient. A SchemaRDD is like relational database table consisting of columns and rows. For that reason we tend to use our own implementation, which is a type safe directed linear pipeline without any type restrictions. The architectural decision whether to use an alpha standard from the MLlib or to use the own implementation is not made yet. Possible libraries to utilise are the following.

MLlib for decision modelling - included in Spark

MLlib Pipelines for modular pipeline design - alpha version included in Spark since 1.2

GraphX for clustering - included in Spark

Sringmetric for similarity calculation - Open source Scala library [Str14]

Weka (optional) for decision modelling and preprocessing - Open source Java machine learning software [Wek14]

Apache Mahout (optional) for decision modelling and preprocessing - Open source Hadoop machine learning library [Mah14]

Besides its output value every pipe in the pipeline produces an evaluation output. Basically there are three input sources an evaluation of a single pipe is based on. These are the corpus containing all tuples, the gold standard containing all true duplicate pairs and the output of the analysed pipe. Currently implemented evaluation measures are set sizes, relative errors, recall and precision. Evaluation is also a performance critical task because it involves expensive calculations. An evaluation log output example of the actual prototype can be seen in Figure 2.

Implementing a wide range of algorithms is not planned. The focus lies on extensibility. There are two different blocking algorithms implemented so far. These are Sorted Neighbourhood [Chr12b] and Suffix Array Blocking [Chr12b]. For clustering there is a base class with functionality to derive clustering algorithms from. The only available implementation at the moment is a transitive closure clustering.

5 Conclusion and Future work

At the moment there is no open source big data deduplication framework available which is scalable with respect to the amount cluster nodes. The briefly introduced deduplication framework called DduP tries to close this gap. It includes pipelining interfaces and implementations for every step mentioned in the overall process. It offers only a small variety of algorithms but aims to be extensible. First tests of the framework are promising but further development and evaluation still needs to be done.

Next step will be the evaluation of the framework. Evaluation will cover different amounts of data and different cluster settings to evaluate how the process scales. Possible cluster settings are 2, 4 and 8 worker nodes plus one master. The main evaluation criteria is the runtime of the process. Performance measures like recall and precision are not that relevant because algorithm performance is not in the focus.

A difficult task is to provide enough data to saturate the cluster. There are two different ways to get test data. The first one is to generate synthetic data and the second one is to use real world data with real duplicates. The usage of real data is difficult, because the gold standard is typically unknown. Combining the two approaches seems to be promising. That means taking a real set of data and randomly duplicate and falsify some tuples by imitating human behaviour. Introducing typos or mistakes based on phonetic equality are only two possibilities.

A set of 10^6 tuples and a size of about 100 MB seems to be a good amount of input data to saturate a small cluster. Depending on the reduction ratio of the search space reduction step this amount of data can easily lead to a very long runtime. Obtaining such a huge set of clean real data is also difficult. In 2014 there were less than $5 \cdot 10^6$ articles in the English Wikipedia [Wik15]. For our purpose data tuples following a scheme are needed and that are much less than $5 \cdot 10^6$ articles. One possible data source could be the Musicbrainz.org

```
##### READING ##############################################
# Number of Tuples:                                  13555 #
#-------------------------------------------------------#
# Runtime Reading:                       00d 00:00:00.293 #
# Runtime Analyser:                      00d 00:00:01.602 #
#########################################################
##### GOLD STANDARD ########################################
# Number of Pairs:                                    5754 #
#-------------------------------------------------------#
# Runtime Gold Standard:                 00d 00:00:00.059 #
# Runtime Analyser:                      00d 00:00:02.490 #
#########################################################
##### BLOCKING #############################################
# Reduced search space size:                         14200 #
# Naive search space size:                        91862235 #
# Reduction ratio:                           0,999845421 #
# Max possible recall:                       0,625825513 #
#-------------------------------------------------------#
# Runtime Blocking:                      00d 00:00:00.055 #
# Runtime Analyser:                      00d 00:00:04.350 #
#########################################################
##### TRAINING SET BUILDING ################################
# Training and test set size:                          200 #
#-------------------------------------------------------#
# Runtime Training Set Building:         00d 00:00:00.439 #
# Runtime Analyser:                      00d 00:00:00.188 #
#########################################################
##### DECISSION TRAINING ###################################
# Training set size:                                   121 #
# Test set size:                                        79 #
# Training Error:                            0,075949367 #
#-------------------------------------------------------#
# Runtime Decission Training:            00d 00:00:01.160 #
# Runtime Analyser:                      00d 00:00:00.457 #
#########################################################
##### DECISSION ############################################
# Number of duplicate pairs found:                     9809 #
#-------------------------------------------------------#
# Runtime Decission:                     00d 00:00:00.015 #
# Runtime Analyser:                      00d 00:00:00.775 #
#########################################################
##### CLUSTERING ###########################################
# Recall:                                    0,551963851 #
# Precission:                                0,341578834 #
#-------------------------------------------------------#
# Number of clusters:                                  2865 #
# Average cluster size:                      2,769633508 #
#-------------------------------------------------------#
# Runtime Clustering:                    00d 00:00:04.837 #
# Runtime Analyser:                      00d 00:00:01.993 #
#########################################################
```

Figure 2: Sample log output of the current prototype detecting duplicates in a small example data set. Due to Sparks lazy evaluation the runtimes are not always correctly separated into pipe and analyser runtime.

[Mus14] database which is a free music database. The database has a complex scheme and contains many informations about released music albums. The first application of the DduP framework could be the deduplication of the Musicbrainz database.

Another approach to get input data seems to be sensible. Taking a set of clean real input data for each feature dimension and do a cartesian product over all dimensions. Having n feature dimensions with k different real data values each, the resulting test set size is k^n. That means with 4 feature dimensions and 32 different values our target of 10^6 is reached. This theoretical data set is not usable because there are many tuples only varying in one dimension what would be a duplicate. Increasing the size of dimensions and data values per dimension solves this problem.

For now there is only one very old tool called "DbGen" [DbG14] from the University Texas in Austin which can do this kind of data generation. It produces clusters of falsified person tuples. Falsification and other parameters can be adjusted. Basis for the generation are a set of 62491 different names and 42115 different zip codes. Addresses and social security numbers gets generated. It looks like we have to implement own tools to prepare test data.

References

[Chr12a] Peter Christen. A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. *Knowledge and Data Engineering, IEEE Transactions on*, 24(9):1537–1555, Sept 2012.

[Chr12b] Peter Christen. Data matching. *Data-Centric Systems and Appl., Springer*, 2012.

[DbG14] DbGen. `http://www.cs.utexas.edu/users/ml/riddle/index.html`, 2014.

[Had14] Apache Hadoop. `http://hadoop.apache.org`, 2014.

[KTR12] Lars Kolb, Andreas Thor, and Erhard Rahm. Dedoop: Efficient Deduplication with Hadoop. *PVLDB*, 5(12):1878–1881, 2012.

[Mah14] Apache Mahout. `http://mahout.apache.org`, 2014.

[MLl14] Apache Spark MLlib. `https://spark.apache.org/mllib`, 2014.

[Mus14] Musicbrainz. `https://musicbrainz.org/`, 2014.

[Pip14] Tinkerpop Pipes. `http://www.tinkerpop.com`, 2014.

[Str14] Stringmetric. `https://github.com/rockymadden/stringmetric`, 2014.

[TLX13] Yufei Tao, Wenqing Lin, and Xiaokui Xiao. Minimal mapreduce algorithms. In *Proceedings of the 2013 international conference on Management of data*, pages 529–540. ACM, 2013.

[Wek14] Weka. `http://www.cs.waikato.ac.nz/ml/weka`, 2014.

[Wik15] Wikipedia. `https://en.wikipedia.org/wiki/Wikipedia:Size\_of\_Wikipedia`, 01 2015.

[XCD+14] Reynold S Xin, Daniel Crankshaw, Ankur Dave, Joseph E Gonzalez, Michael J Franklin, and Ion Stoica. GraphX: Unifying Data-Parallel and Graph-Parallel Analytics. *arXiv preprint arXiv:1402.2394*, 2014.

[ZCF+10] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 10–10, 2010.

Tutorienprogramm

Distance-based Multimedia Indexing

Christian Beecks Merih Seran Uysal Thomas Seidl

Data Management and Data Exploration Group
RWTH Aachen University
Germany
{beecks, uysal, seidl}@cs.rwth-aachen.de

Abstract: This tutorial aims at providing a unified and comprehensive overview of the state-of-the-art approaches to distance-based multimedia indexing.

1 Introduction

Concomitant with the explosive growth of the digital universe [GCM+08], an immensely increasing amount of multimedia data is generated, processed, and finally stored in very large multimedia databases. The rapid expansion of the internet and the extensive spread of mobile devices allow users to generate and share multimedia data everywhere and at any time. As a result, multimedia databases tend to grow continuously without any restriction and are thus no longer manually manageable by humans. Automatic approaches that allow for effective and efficient information access to massive multimedia databases become immensely important.

Multimedia retrieval approaches [LSDJ06] are one class of information access approaches that allow to manage and access multimedia databases with respect to the users' information needs. These approaches deal with the representation, storage, organization of, and access to information items [BYRN11]. In fact, they can be thought of approaches allowing users to *search*, *browse*, *explore*, and *analyze* multimedia databases by means of similarity relations among multimedia objects.

One promising and widespread approach to define similarity between multimedia objects consists in automatically extracting inherent properties of multimedia objects and comparing them with each other. For this purpose, the content-based properties of multimedia objects are modeled by feature representations which are comparable by means of *distance-based similarity measures*. This class of similarity measures follows a rigorous mathematical interpretation [She57] and allows domain experts and database experts to address the issues of effectiveness and efficiency simultaneously and independently. In fact, it has become mandatory for current distance-based similarity measures to be indexable in order to facilitate large-scale applicability.

2 Tutorial Outline

This tutorial aims at providing a unified and comprehensive overview of the state-of-the-art approaches to distance-based multimedia indexing. We intend to cover a broad target audience starting from beginners to experts in the domain of multimedia databases.

The tutorial is structured into four parts as shown below:

- Object Representation
  - Feature Extraction
  - Feature Representations
  - Algebraic Properties
  - Clustering-based Computation
- Fundamental Similarity Models
  - Similarity Measures
  - Dissimilarity Measures
- Efficient Query Processing
  - Similarity Queries
  - Lower-Bounding
- Indexing
  - Spatial Indexing
  - High-dimensional Indexing
  - Metric and Ptolemaic Indexing

3 About The Presenters

Christian Beecks is a postdoctoral researcher in the data management and data exploration group at RWTH Aachen University, Germany. His research interests include efficient content-based multimedia retrieval and exploration, adaptive distance-based similarity measures such as the Earth Mover's Distance [RTG00], Signature Quadratic Form Distance [BUS10], and Signature Matching Distance [BKS13], as well as metric and Ptolemaic indexing.

Merih Seran Uysal is a researcher in the data management and data exploration group at RWTH Aachen University, Germany. Her research interests include similarity search in multimedia databases and efficient query processing based on adaptive distance-based similarity measures.

Thomas Seidl is a professor for computer science and head of the data management and data exploration group at RWTH Aachen University, Germany. His research interests include data mining and database technology for multimedia and spatio-temporal databases in engineering, communication and life science applications. Prof. Seidl received his Diplom (MSc) in 1992 from TU Muenchen and his PhD (1997) and venia legendi (2001) from LMU Muenchen.

4 Acknowledgments

This work is funded by DFG grant SE 1039/7-1. It is partly based on the work of Beecks [Bee13].

References

[Bee13] Christian Beecks. *Distance-based similarity models for content-based multimedia retrieval*. PhD thesis, RWTH Aachen University, 2013.

[BKS13] Christian Beecks, Steffen Kirchhoff, and Thomas Seidl. Signature Matching Distance for Content-based Image Retrieval. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, pages 41–48, 2013.

[BUS10] Christian Beecks, Merih Seran Uysal, and Thomas Seidl. Signature Quadratic Form Distance. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 438–445, 2010.

[BYRN11] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley Professional, 2011.

[GCM+08] John F. Gantz, Christopher Chute, Alex Manfrediz, Stephen Minton, David Reinsel, Wolfgang Schlichting, and Anna Toncheva. The Diverse and Exploding Digital Universe. *IDC White Paper*, 2, 2008.

[LSDJ06] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2(1):1–19, 2006.

[RTG00] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[She57] Roger N Shepard. Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22:325–345, 1957.

Many-Core-Architekturen zur Datenbankbeschleunigung – Tutorial –

Kai-Uwe Sattler[1], Jens Teubner[2], Felix Beier[1], Sebastian Breß[2]

[1] Technische Universität Ilmenau, Ilmenau, Germany
{first.last}@tu-ilmenau.de
[2] Technische Universität Dortmund, Dortmund, Germany
{first.last}@cs.tu-dortmund.de

Physikalische und technologische Grenzen bei der Erhöhung der Taktfrequenz von Prozessoren haben in den letzten Jahren die Entwicklung von Multi- und Many-Core-Architekturen forciert. Die Ausnutzung dieser Architekturen erfordert jedoch eine weitgehende Parallelisierung von Berechnungen. Für den Datenbankbereich bedeutet dies einerseits ein Überdenken etablierter Datenstrukturen und Verfahren, eröffnet aber gleichzeitig neue Möglichkeiten der Beschleunigung und Skalierung der Datenbankverarbeitung. Ziel des Tutoriums ist es daher, einen Überblick über den Stand der Forschung und die Einsatzmöglichkeiten von Many-Core-Architekturen in Datenbanksystemen zu geben. Neben Standard-Prozessoren stehen dabei insbesondere GPGPU-Architekturen im Mittelpunkt, die schon heute die Nutzung von Tausenden Cores ermöglichen. Ausgehend von einer Vorstellung aktueller Many-Core- und GPU-Architekturen sowie der darauf basierenden Programmiermodelle und APIs werden eine Reihe repräsentativer Einsatzfelder diskutiert. Neben Standard-Datenbankoperatoren, Indexierung von Daten sowie der Realisierung von benutzerdefinierten Funktionen (UDFs) werden auch Möglichkeiten der Unterstützung von DBMS-Komponenten wie dem Optimierer behandelt. Einen zweiten Schwerpunkt bilden Aspekte heterogener Plattformen aus CPU und GPU wie Fragen der Platzierung und des Datentransfers. Zielgruppe des Tutoriums sind Datenbank-Forscher und Entwickler, die sich für den Einsatz moderner Hardwareplattformen in Datenbanksystemen interessieren und sich über den Stand der Forschung informieren möchten. Sie erhalten einen Einblick in Programmiermodelle und -techniken sowie Entscheidungsunterstützung beim Einsatz von Many-Core- und GPU-Architekturen zur Datenbankbeschleunigung.

Kai-Uwe Sattler ist Professor für Datenbanken und Informationssysteme und leitet das gleichnamige Fachgebiet an der TU Ilmenau. Er promovierte 1998 an der Universität Magdeburg und habilitierte sich 2003 an der gleichen Universität. Zu seinen Forschungsgebieten gehören Anfrageverarbeitung mit Verteilungs- und Parallelisierungsaspekten, Datenstrommanagement sowie der Einsatz modernen Hardwarearchitekturen. Er ist Koautor mehrerer Lehrbücher zu Datenbankkonzepten sowie Algorithmen und Datenstrukturen.

Jens Teubner leitet seit April 2013 den Lehrstuhl für Datenbanken und Informationssysteme an der TU Dortmund. Zuvor forschte er als Postdoktorand bzw. Oberassistent am

IBM T.J. Watson Forschungslabor in den USA sowie in der Systems Group an der ETH Zürich. Er promovierte 2006 an der TU München bei Prof. Dr. Torsten Grust auf dem Gebiet der XML-Verarbeitung und diplomierte an der Fakultät für Physik der Universität Konstanz bei Prof. Dr. Ernst Bucher.

Forschungsschwerpunkt von Jens Teubner ist die Implementierung von Datenbanksystemen auf modernen Rechnerarchitekturen. Das von ihm gegründete Forschungsprojekt Avalanche an der ETH Zürich wurde schnell zur international führenden Gruppe was die Verwendung von Field-Programmable Gate Arrays (FPGAs) für die Beschleunigung von Datenbankaufgaben anbelangt. Er ist Mit-Entwickler der derzeit schnellsten Join-Implementation für moderne Hauptspeicher-Datenbanksysteme und zeigte, wie sich Netzwerkkarten mit Hardwarebeschleunigung (sog. Remote Direct Memory Access, RDMA) besonders effizient für verteilte Datenbanken einsetzen lassen. Im Rahmen seiner Promotion beschäftigte Jens Teubner sich außerdem mit der hoch-skalierbaren Verarbeitung von XML, insbesondere unter Verwendung relationaler Datenbank-Back-Ends. Das Ergebnis seiner Promotion – der XQuery-Compiler Pathfinder – ist heute Teil des Open Source-Systems MonetDB/XQuery.

Jens Teubner erhielt für seine Dissertation Pathfinder: XQuery Compilation Techniques for Relational Database Targets im März 2007 den Dissertationspreis der Gesellschaft für Informatik, Fachgruppe Datenbanken (BTW-Dissertationspreis). Außerdem erhielt er einen Best Demo Award auf der BTW 2011, einen Best Paper Award auf der FCCM 2013 sowie ein Honorable Mention auf der ICDE 2013.

Felix Beier hat Informatik an der Technischen Universität Ilmenau studiert und promoviert dort seit 2010 am Fachgebiet Datenbanken und Informationssysteme. Sein Forschungsschwerpunkt ist die Parallelisierung von Indexoperationen auf hybriden CPU/GPU Systemen. 2013 wurde er für seine Forschungsarbeit mit dem IBM PhD Fellowship Award ausgezeichnet. Neben parallelen Algorithmen für moderne Prozessorarchitekturen gehören Datenstrommanagementsysteme und die Programmierung mit C++ zu seinen Interessen.

Sebastian Breß erhielt 2010 einen Bachelorabschluss und 2012 einen Masterabschluss in Informatik an der Universität Magdeburg. Aktuell arbeitet er an der Technischen Universität Dortmund und erforscht effiziente Techniken zur Anfrageoptimierung in heterogenen Prozessorsystemen. Dabei liegt der Fokus auf Ansätzen die keine Details der Prozessorarchitektur für die Optimierung benötigen. Während seiner Arbeit hat er zwei Systeme entwickelt: 1) Den lernenden Anfrageoptimierer HyPE und 2) CoGaDB, eine GPU-beschleunigte Hauptspeicherdatenbank für OLAP Szenarien. Er organisierte das „Deutsche Community Treffen für GPUs in Datenbanken“ 2014 in Dortmund und ist Mitorganisator des „International Workshop on Data (Co-)Processing on Heterogeneous Hardware“ (DAPHNE) bei der EDBT/ICDT 2015.

Skalierbare NoSQL- und Cloud-Datenbanken in Forschung und Praxis

Felix Gessert, Norbert Ritter

Database and Information Systems Group
Universität of Hamburg
Vogt-Kölln Straße 33
22527 Hamburg, Deutschland
{gessert, ritter}@informatik.uni-hamburg.de

Abstract: Die rasante Entwicklung nicht-relationaler, verteilter NoSQL Datenbanksysteme hat in den letzten Jahren einen beispiellosen Aufschwung erlebt. Zwei zentrale Probleme haben diesen Prozess angestoßen: die gewaltigen Mengen von „User-generated-Content“ in modernen Anwendungen und die damit einhergehenden Anfragelasten und Datenvolumnia, sowie der Ruf von Entwicklern nach problemspezifischen Datenmodellen und Schemaflexibilität. Das Tutorium „Skalierbare NoSQL- und Cloud-Datenbanken in Forschung und Praxis“ bietet einen umfassenden Überblick über die Konzepte und Techniken der relevantesten NoSQL- und Cloud-Datenbanken, mit einem besonderen Fokus auf Trade-Offs im Bereich Skalierbarkeit, Konsistenz und Anfragemächtigkeit. Es werden sowohl zugrundeliegende theoretische Konzepte, bahnbrechende wissenschaftliche Beiträge, als auch praktische Aspekte wie APIs, Lizenz- und Pricingmodelle sowie Architekturen diskutiert.

1 Beschreibung

Für einen breiten Überblick beginnt das Tutorium mit einer Motivation der NoSQL-Bewegung [SF12], sowie einer Kategorisierung von NoSQL-Systemen anhand ihres Datenmodells. Anschließend werden die wichtigen theoretischen Grundlagen eingeführt, wie beispielsweise Replikation, das CAP-Theorem [GL02] und Consistent Hashing. Diese Techniken sind die Grundlagen dafür, die Garantien und Fähigkeiten der anschließend detailliert beleuchteten Systeme beurteilen zu können. Zu den behandelten Systemen zählen u.a. die Dokumentendatenbank MongoDB, die Wide-Column Stores HBase und Cassandra, sowie die Key-Value Stores Redis und Riak. Anhand der Verteilungsarchitektur dieser Systeme mit ihren speziellen Replikations- und Partitionierungsmechanismen werden die erzielbaren Konsistenz- und Skalierbarkeitseigenschaften abgeleitet [FWGR14]. Um ein Verständnis für die jeweiligen Sweetspots im praktischen Einsatz zu erlangen, werden anschließend die Datenmodellierung und die Anfrageschnittstellen behandelt. Im Fokus stehen hier das Konzept der Schemalosigkeit und seine Implikationen, Key-Design, Schlüsselwahl bei Range- und Hash-Partitionierung, Query-Sprachen und Denormalisierungsstrategien.

Da Big Data Analytics und Data Science ein zunehmend wichtiges und stark diskutiertes Thema sind, werden auch Architekturen und Schnittstellen zur Integration operativer NoSQL-Datenbanken mit analytischen Systemen wie Hadoop und Spark diskutiert. Hierbei soll klar herausgearbeitet werden, dass Big Data zwei Ausprägungen hat: Data Management und Data Analytics, die im Kern jedoch auf denselben Abstraktionen aufbauen, um Skalierbarkeit zu erzielen und denselben Verfügbarkeits-Konsistenz-Trade-Offs unterworfen sind. Als Leitfaden und Orientierungshilfe werden die diskutierten NoSQL Datenbanken abschließend bezüglich mehrerer Eigenschaften wie Konsistenz, Anfragemächtigkeit und Lernkurve eingeordnet.

Der darauffolgende Abschnitt beschäftigt sich mit Cloud-Datenbanken [LS13] in seinen verschiedenen Ausprägungsformen: Cloud-hosted databases, Database-as-a-Service (DBaaS) [GBR14] und Backend-as-a-Service (BaaS) [GFW+14]. Einleitend wird die Multi-Tenancy-Problematik diskutiert, sowie das Konzept der Service-Levels-Agreements und des Workload-Managements eingeführt. Anhand repräsentativer Vertreter der jeweiligen Cloud-Datenbank-Kategorien (z.B. dem Amazon Relational Database Service als Beispiel für ein relationales DBaaS) soll hier ein praktischer Einblick in die konkrete Arbeit mit diesen Diensten gegeben werden. Da ein Großteil aller Cloud-Datenbank proprietärer Natur ist, müssen Rückschlüsse über nichtfunktionale Eigenschaften i.d.R. über vorhandene wissenschaftliche Publikationen (z.B. MegaStore [BBC+11] als Grundlage von Googles DataStore) angestellt werden. Hierbei werden wir zeigen, dass viele der Systeme dieselben Architekturen und Trade-Offs wie die zuvor diskutieren Open-Source NoSQL-Systeme aufweisen.

Als ein Beispiel dafür, wie NoSQL- und Cloud-Datenbanken auch althergebrachte Herangehensweisen in der Softwareentwicklung in Frage stellen, werden wird das Backend-as-a-Service Paradigma diskutieren, das es sich zum Ziel gemacht hat, die Web und Appentwicklung durch eine Kombination aus standardisierter Backendlogik und Cloud-Datenbanken radikal zu vereinfachen. Das Tutorium schließt mit einer Zusammenfassung und einer Aufstellung der wichtigsten offenen Forschungsfragen für NoSQL- und Cloud-Datenbanken.

2 Zielgruppe

Das Tutorium richtet sich an Anwender (z.B. Softwareentwickler und -Architekten) und Wissenschaftler, die ihre Kenntnisse im Bereich NoSQL- und Cloud-Datenbanken vertiefen oder sich über die wichtigsten offenen Probleme informieren möchten. Vorkenntnisse in den Bereichen Datenbanksysteme, verteilte Systeme und Softwareentwicklung sind von Vorteil, aber nicht zwingend erforderlich.

3 Organisatoren

Felix Gessert ist Doktorand an der Gruppe Datenbanken und Informationssysteme der Universität Hamburg. Er forscht in den Bereichen skalierbare Datenbanksysteme, Transaktionen, Cloud Computing und Web-Technologien. Sein besonderer Fokus gilt dem Caching und der Transaktionsverarbeitung im Kontext von Cloud-Datenbanksystemen. Er ist außerdem Gründer des Spin-Offs Baqend, das die Forschungsergebnisse in Form einer Backend-as-a-Service Plattform umsetzt.

Norbert Ritter ist Professor für Informatik und Leiter der Gruppe Datenbanken und Informationssysteme. Seine Forschungsinteressen umfassen u.a. verteilte und föderierte Datenbanksysteme, Transaktionsverarbeitung, Caching, Cloud Data Management, Informationsintegration und autonome Datenbanksysteme.

Literatur

[BBC+11] Jason Baker, Chris Bond, James C. Corbett, J. J. Furman, Andrey Khorlin, James Larson, Jean-Michel Léon, Yawei Li, Alexander Lloyd und Vadim Yushprakh. Megastore: Providing Scalable, Highly Available Storage for Interactive Services. In *CIDR*, Jgg. 11, Seiten 223–234, 2011.

[FWGR14] Steffen Friedrich, Wolfram Wingerath, Felix Gessert und Norbert Ritter. NoSQL OLTP Benchmarking: A Survey. In Erhard Plödereder, Lars Grunske, Eric Schneider und Dominik Ull, Hrsg., *44. Jahrestagung der Gesellschaft für Informatik, Informatik 2014, Big Data - Komplexität meistern, 22.-26. September 2014 in Stuttgart, Deutschland*, Jgg. 232 of *LNI*, Seiten 693–704. GI, 2014.

[GBR14] Felix Gessert, Florian Bucklers und Norbert Ritter. Orestes: A scalable Database-as-a-Service architecture for low latency. In *Workshops Proceedings of the 30th International Conference on Data Engineering Workshops, ICDE 2014, Chicago, IL, USA, March 31 - April 4, 2014*, Seiten 215–222. IEEE, 2014.

[GFW+14] Felix Gessert, Steffen Friedrich, Wolfram Wingerath, Michael Schaarschmidt und Norbert Ritter. Towards a Scalable and Unified REST API for Cloud Data Stores. In Erhard Plödereder, Lars Grunske, Eric Schneider und Dominik Ull, Hrsg., *44. Jahrestagung der Gesellschaft für Informatik, Informatik 2014, Big Data - Komplexität meistern, 22.-26. September 2014 in Stuttgart, Deutschland*, Jgg. 232 of *LNI*, Seiten 723–734. GI, 2014.

[GL02] Seth Gilbert und Nancy A. Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, 2002.

[LS13] Wolfgang Lehner und Kai-Uwe Sattler. *Web-Scale Data Management for the Cloud*. Springer, 2013.

[SF12] Pramod J. Sadalage und Martin Fowler. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education, 2012.

Big-Data-Anwendungsentwicklung mit SQL und NoSQL

Jens Albrecht
Technische Hochschule Nürnberg
jens.albrecht@th-nuernberg.de

Uta Störl
Hochschule Darmstadt
uta.stoerl@h-da.de

Überblick

Bei der Verarbeitung und Auswertung von Big Data stoßen klassische relationale Datenbanken an ihre Grenzen. So ist die Speicherung von Daten im Multi-Terabyte-Bereich damit zwar möglich, aber in der Regel aufgrund der Lizenz- und Speicherkosten unwirtschaftlich. Datenanalysen, bei denen große Datenbereiche gelesen werden müssen, erfordern darüber hinaus cluster-fähige Systeme für die Skalierung.

Neben dem Datenvolumen stellen die Vielfalt der Daten und die Dynamik von Schemaänderungen ein Problem für relationale Datenbanken dar. Insbesondere Web-Anwendungen sind, neben hohem Datendurchsatz, durch sehr kurze Release-Zyklen geprägt. Schemaänderungen müssen im laufenden Betrieb durchgeführt werden, da Downtimes meist inakzeptabel sind.

Um die genannten Schwachstellen zu adressieren, haben sich in den letzten Jahren einige neue Datenbank-Ansätze etabliert. Zu nennen sind hier zum einen NoSQL-Datenbanken, die sich in der Regel durch ein sehr einfaches, aber flexibles Datenmodell und sehr gute Skalierbarkeit auszeichnen. Zum anderen hat sich für die Speicherung großer Datenmengen sowie deren Aufbereitung und Analyse das Apache-Framework Hadoop als Standard etabliert.

Das Tutorium legt den Schwerpunkt auf zwei relativ neue Aspekte dieser Technologien, die jeweils in einem 90-minütigen Vortrag vorgestellt werden:

1. Datenanalyse mit SQL auf Hadoop
2. NoSQL-Anwendungsentwicklung mit Objekt-NoSQL Mappern

Im Folgenden werden diese Themen kurz vorgestellt.

Datenanalyse mit SQL auf Hadoop

Hadoop ist ein Framework zur Speicherung und Verarbeitung großer Datenmengen in einem Cluster aus Standard-Servern. Es besteht im Kern aus dem verteilten Hadoop-

Filesystem sowie dem Map-Reduce-Framework. Letzteres stellt eine Ablaufumgebung für die verteilte Ausführung von Jobs, die in Java programmiert werden müssen, dar.

Die Programmierung von Map-Reduce-Jobs für die Datenanalyse ist, verglichen mit SQL, sehr aufwändig und das Know-How dafür in den wenigsten Firmen vorhanden. Andererseits sind auch bei Big Data viele Abfragen inhaltlich wenig komplex, wie folgende Beispiele illustrieren:

- Ermittlung der Häufigkeiten der Seitenaufrufe (Visits) für bestimmte Web-Sites
- Ermittlung durchschnittlicher Werte von Sensordaten pro Messstation
- Suche nach bestimmten Hash-Tags in Twitter-Nachrichten

Diese und ähnliche Abfragen lassen sich sehr einfach mit SQL ausdrücken. Notwendig dafür ist ein relationales Datenmodell, das für viele Daten, die satzweise erzeugt werden und eine gewisse Struktur aufweisen, einfach definiert werden kann. Eine verteilte SQL Execution Engine sorgt dann für die Ausführung im Hadoop-Cluster.

Bereits im Jahr 2007 wurde zu diesem Zweck Hive entwickelt, eine SQL-Engine, die Map-Reduce-Jobs auf Hadoop generiert. In den vergangenen zwei Jahren ist eine Vielzahl weiterer SQL-Systeme für Hadoop auf den Markt gekommen. Anders als Hive setzen Produkte wie Cloudera Impala oder Pivotal Hawk auf eigene Execution Engines anstelle Map-Reduce, um eine deutlich bessere Performance zu erreichen. Parallel dazu arbeiten die großen Datenbank-Hersteller Oracle, IBM und Microsoft an Lösungen, um Hadoop bestmöglich in ihre Produkte zu integrieren. Ganz neue Möglichkeiten eröffnet Apache Spark, ein In-Memory-Framework zur Datenverarbeitung. Mit Apache Spark SQL gibt es auch dafür bereits einen SQL-Aufsatz.

Der Vortrag gibt einen Überblick über existierende SQL-Zugriffsmöglichkeiten auf Daten in Hadoop und stellt die verschiedenen Architekturansätze anhand einer Klassifikation gegenüber.

NoSQL-Anwendungsentwicklung mit Objekt-NoSQL Mappern

Bei der Anwendungsentwicklung für NoSQL-Datenbanken gibt es verschiedene Herausforderungen: Da sich die Anfragesprachen für NoSQL-Systeme erheblich unterscheiden, führt eine native Anwendungsentwicklung unmittelbar zu starken Abhängigkeiten von einem spezifischen System. Außerdem führt die Schemaflexibilität der NoSQL-Datenbanken dazu, dass die Verantwortung für das Schema-Management in die Anwendung verlagert wird.

Einen möglichen Lösungsansatz für diese Herausforderungen stellen Objekt-NoSQL Mapper dar. Diese Systeme übertragen die Idee der Objekt-Relationalen Mapper auf NoSQL-Systeme. Aktuell finden sich am Markt dabei sowohl Produkte aus dem Bereich der Objekt-Relationalen Mapper, welche um die Unterstützung für (einige) NoSQL-Systeme erweitert wurden (beispielsweise EclipseLink, Hibernate und DataNucleus) als auch speziell

für den Bereich des Objekt-NoSQL Mapping entwickelte Systeme wie Kundera. Sofern es sich um Objekt-NoSQL Mapper für Java handelt, setzen fast alle diese Produkte auf die Java Persistence Language (JPQL) aus der Java Persistence API (JPA) als einheitliche Anfragesprache.

Aufgrund der größeren Flexibilität der Abbildung eines Objektmodells auf NoSQL-Datenmodelle und der sehr unterschiedlichen Mächtigkeit der NoSQL-Anfragesprachen unterscheiden sich die existierenden Produkte derzeit allerdings sowohl konzeptionell als auch bezüglich der Unterstützung verschiedener NoSQL-Systeme erheblich.

Im Tutorium wird ein Überblick über die verschiedenen existierenden Ansätze gegeben, diese werden kategorisiert und miteinander verglichen. Eine weitere wichtige Frage beim Einsatz von Mappern ist die Auswirkung auf die Performance der Lese- und Schreibzugriffe. Auch hierfür werden im Tutorium Untersuchungen präsentiert und Einsatzempfehlungen für Objekt-NoSQL Mapper gegeben.

Zielgruppe

Zielgruppe des Tutoriums sind Praktiker und Anwender, die sich über verschiedene Architekturkonzepte und den aktuellen Stand in Produkten zur Big-Data-Anwendungsentwicklung informieren wollen.

Die Vortragenden

Jens Albrecht ist seit 2014 Professor für Datenbanken und Big Data an der Technischen Hochschule Nürnberg, nachdem er bereits zwei Jahre an der Hochschule Würzburg-Schweinfurt eine Professur inne hatte. Zuvor war er als Berater bei Oracle und als WH-Architekt in der GfK in Nürnberg tätig. Sein besonderes Interesse gilt Technologien zur Aufbereitung und Analyse großer Datenmengen.

Uta Störl ist seit 2005 Professorin für Datenbanken und Informationssysteme an der Hochschule Darmstadt. Aktuelle Schwerpunkte ihrer Arbeit sind Technologien für NoSQL-Datenbanksysteme sowie interdisziplinäre Projekte im Bereich Digital Humanities. Nach ihrer Promotion 1999 an der Friedrich-Schiller-Universität in Jena hat sie im Research- bzw. Projektmanagement-Bereich der Dresdner Bank in Frankfurt am Main gearbeitet.

GI-Edition Lecture Notes in Informatics

P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.

P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.

P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Lan-guage to Information Systems, NLDB'2001.

P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensor-gestützte Chirurgie; Workshop des SFB 414.

P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.

P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hoch-schule, UH'2001.

P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.

P-8 Reinhard Keil-Slawik, Johannes Magen-heim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.

P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeits-tagung.

P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Man-agement: Sharing Experiences about the Sharing Experience.

P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.

P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.

P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.

P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Se-mantic Web – XSW 2002.

P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.

P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Busi-ness 2002.

P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletz-barkeit meistern, 16. DFN Arbeitstagung.

P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informations-systeme – MobIS 2002.

P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Infor-matik 2002 – 32. Jahrestagung der Gesell-schaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund.

P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Infor-matik 2002 – 32. Jahrestagung der Gesell-schaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund (Ergänzungs-band).

P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Metho-den und Werkzeuge für die Entwicklung von Informationssystemen.

P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.

P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit

P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middle-ware für XML-Anwendungen

P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003

P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Daten-banksysteme für Business, Technologie und Web

P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin

P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Er-fahrungen und Visionen

P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems

P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications

P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003

P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft

P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)

P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)

P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik

P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik

P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003

P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003

P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004

P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing

P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste

P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistance, Scalability, Transactions – Database Mechanisms for Mobile Applications

P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services

P-45 Bernhard Rumpe, Wofgang Hesse (Hrsg.): Modellierung 2004

P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment

P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society

P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications

P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven

P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik

P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004

P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration

P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Adhoc-Netze Medienintegration

P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government

P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen

P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality

P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments

P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für inforrmatische Bildung

P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen

P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit

P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications

P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005

P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik

P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)

P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)

P-69 Robert Hirschfeld, Ryszard Kowalcyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005

P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)

P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005

P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment

P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): "Heute schon das Morgen sehen"

P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology

P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture

P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz

P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

P-78 K.-O. Wenkel, P. Wagner, M. Morgenstern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel

P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006

P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce

P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS´06

P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006

P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics

P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications

P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems

P-86 Robert Krimmer (Ed.): Electronic Voting 2006

P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik

P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006

P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur

P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006

P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006

P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1

P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2

P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen

P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies

P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices

P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik

P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik

P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten

P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments

P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)

P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme

P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007

P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops

P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications

P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures

P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1

P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2

P-111 Christian Eibl, Johannes Magenheim, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik

P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis

P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)

P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics

P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007

P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007

P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and manegement The 4th International Conference on Grid Service Engineering and Management GSEM 2007

P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007

P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications

P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges

P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik

P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)

P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Speciel Interest Groups PARS and PARVA

P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT: Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung

P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)

P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008

P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany

P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)
Sigsand-Europe 2008
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems

P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
1. DFN-Forum Kommunikations-technologien Beiträge der Fachtagung

P-131 Robert Krimmer, Rüdiger Grimm (Eds.)
3rd International Conference on Electronic Voting 2008
Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting. CC

P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik

P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 1

P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 2

P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik – Aktuelle Forschungsergebnisse

P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics
GCB 2008

P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures

P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008

P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der Anwendungsentwick-lung und -wartung im Fachbereich Wirtschaftsinformatik

P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)
IMF 2008 - IT Incident Management & IT Forensics

P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)
Modellierung betrieblicher Informations-systeme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management

P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung

P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)

P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management

P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)

P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009

P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I2CS 2009

P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
2. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)
Software Engineering 2009 - Workshopband

P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for Enterprise Software

P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 3rd Int'l Workshop EMISA 2009

P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)
Lernen im Digitalen Zeitalter
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik

P-154 Stefan Fischer, Erik Maehle Rüdiger Reischuk (Hrsg.)
INFORMATIK 2009
Im Focus das Leben

P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2009:
Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«

P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)
German Conference on Bioinformatics 2009

P-158 W. Claupein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)
Precision Agriculture Reloaded – Informationsgestützte Landwirtschaft

P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)
Software Engineering 2010

P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)
Software Engineering 2010 – Workshopband
(inkl. Doktorandensymposium)

P-161 Gregor Engels, Dimitris Karagiannis Heinrich C. Mayr (Hrsg.)
Modellierung 2010

P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)
Vernetzte IT für einen effektiven Staat
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2010

P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)
Mobile und Ubiquitäre Informationssysteme
Technologien, Anwendungen und Dienste zur Unterstützung von mobiler Kollaboration

P-164 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2010: Biometrics and Electronic Signatures Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-165 Gerald Eichler, Peter Kropf, Ulrike Lechner, Phayung Meesad, Herwig Unger (Eds.)
10th International Conference on Innovative Internet Community Systems (I^2CS) – Jubilee Edition 2010 –

P-166 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
3. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-167 Robert Krimmer, Rüdiger Grimm (Eds.)
4th International Conference on Electronic Voting 2010
co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-168 Ira Diethelm, Christina Dörge, Claudia Hildebrandt, Carsten Schulte (Hrsg.)
Didaktik der Informatik
Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik

P-169 Michael Kerres, Nadine Ojstersek Ulrik Schroeder, Ulrich Hoppe (Hrsg.)
DeLFI 2010 - 8. Tagung der Fachgruppe E-Learning der Gesellschaft für Informatik e.V.

P-170 Felix C. Freiling (Hrsg.)
Sicherheit 2010
Sicherheit, Schutz und Zuverlässigkeit

P-171 Werner Esswein, Klaus Turowski, Martin Juhrisch (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2010)
Modellgestütztes Management

P-172 Stefan Klink, Agnes Koschmider Marco Mevius, Andreas Oberweis (Hrsg.)
EMISA 2010
Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme
Beiträge des Workshops der GI-Fachgruppe EMISA (Entwicklungsmethoden für Informationssysteme und deren Anwendung)

P-173 Dietmar Schomburg, Andreas Grote (Eds.)
German Conference on Bioinformatics 2010

P-174 Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)
perspeGKtive 2010
Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“

P-175 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 1

P-176 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 2

P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fähnrich, Bogdan Franczyk, Leszek A. Maciaszek (Eds.)
INFORMATIK 2010
Business Process and Service Science – Proceedings of ISSS and BPSC

P-178 Wolfram Pietsch, Benedikt Krams (Hrsg.)
Vom Projekt zum Produkt
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschafts-informatik (WI-MAW), Aachen, 2010

P-179 Stefan Gruner, Bernhard Rumpe (Eds.)
FM+AM`2010
Second International Workshop on Formal Methods and Agile Methods

P-180 Theo Härder, Wolfgang Lehner, Bernhard Mitschang, Harald Schöning, Holger Schwarz (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW)
14. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS)

P-181 Michael Clasen, Otto Schätzel, Brigitte Theuvsen (Hrsg.)
Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft

P-182 Ronald Maier (Hrsg.)
6th Conference on Professional Knowledge Management
From Knowledge to Action

P-183 Ralf Reussner, Matthias Grund, Andreas Oberweis, Walter Tichy (Hrsg.)
Software Engineering 2011
Fachtagung des GI-Fachbereichs Softwaretechnik

P-184 Ralf Reussner, Alexander Pretschner, Stefan Jähnichen (Hrsg.)
Software Engineering 2011
Workshopband
(inkl. Doktorandensymposium)

P-185 Hagen Höpfner, Günther Specht, Thomas Ritz, Christian Bunse (Hrsg.)
MMS 2011: Mobile und ubiquitäre Informationssysteme Proceedings zur 6. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2011)

P-186 Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.)
11th International Conference on Innovative Internet Community Systems (I^2CS)

P-187 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
4. DFN-Forum Kommunikations-technologien, Beiträge der Fachtagung 20. Juni bis 21. Juni 2011 Bonn

P-188 Holger Rohland, Andrea Kienle, Steffen Friedrich (Hrsg.)
DeLFI 2011 – Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. 5.–8. September 2011, Dresden

P-189 Thomas, Marco (Hrsg.)
Informatik in Bildung und Beruf INFOS 2011
14. GI-Fachtagung Informatik und Schule

P-190 Markus Nüttgens, Oliver Thomas, Barbara Weber (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2011)

P-191 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2011
International Conference of the Biometrics Special Interest Group

P-192 Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.)
INFORMATIK 2011
Informatik schafft Communities

P-193 Wolfgang Lehner, Gunther Piller (Hrsg.)
IMDM 2011

P-194 M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.)
Informationstechnologie für eine nachhaltige Landbewirtschaftung
Fokus Forstwirtschaft

P-195 Neeraj Suri, Michael Waidner (Hrsg.)
Sicherheit 2012
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 6. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)

P-196 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2012
Proceedings of the 11th International Conference of the Biometrics Special Interest Group

P-197 Jörn von Lucke, Christian P. Geiger, Siegfried Kaiser, Erich Schweighofer, Maria A. Wimmer (Hrsg.)
Auf dem Weg zu einer offenen, smarten und vernetzten Verwaltungskultur
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2012

P-198 Stefan Jähnichen, Axel Küpper, Sahin Albayrak (Hrsg.)
Software Engineering 2012
Fachtagung des GI-Fachbereichs Softwaretechnik

P-199 Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.)
Software Engineering 2012
Workshopband

P-200 Gero Mühl, Jan Richling, Andreas Herkersdorf (Hrsg.)
ARCS 2012 Workshops

P-201 Elmar J. Sinz Andy Schürr (Hrsg.)
Modellierung 2012

P-202 Andrea Back, Markus Bick, Martin Breunig, Key Pousttchi, Frédéric Thiesse (Hrsg.)
MMS 2012:Mobile und Ubiquitäre Informationssysteme

P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
5. DFN-Forum Kommunikations-technologien
Beiträge der Fachtagung

P-204 Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.)
12th International Conference on Innovative Internet Community Systems (I2CS 2012)

P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.)
5th International Conference on Electronic Voting 2012 (EVOTE2012)
Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-206 Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)
EMISA 2012
Der Mensch im Zentrum der Modellierung

P-207 Jörg Desel, Jörg M. Haake, Christian Spannagel (Hrsg.)
DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
24.–26. September 2012

P-208 Ursula Goltz, Marcus Magnor, Hans-Jürgen Appelrath, Herbert Matthies, Wolf-Tilo Balke, Lars Wolf (Hrsg.)
INFORMATIK 2012

P-209 Hans Brandt-Pook, André Fleer, Thorsten Spitta, Malte Wattenberg (Hrsg.)
Nachhaltiges Software Management

P-210 Erhard Plödereder, Peter Dencker, Herbert Klenk, Hubert B. Keller, Silke Spitzer (Hrsg.)
Automotive – Safety & Security 2012
Sicherheit und Zuverlässigkeit für automobile Informationstechnik

P-211 M. Clasen, K. C. Kersebaum, A. Meyer-Aurich, B. Theuvsen (Hrsg.)
Massendatenmanagement in der Agrar- und Ernährungswirtschaft
Erhebung - Verarbeitung - Nutzung
Referate der 33. GIL-Jahrestagung
20. – 21. Februar 2013, Potsdam

P-212 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2013
Proceedings of the 12th International Conference of the Biometrics Special Interest Group
04.–06. September 2013
Darmstadt, Germany

P-213 Stefan Kowalewski, Bernhard Rumpe (Hrsg.)
Software Engineering 2013
Fachtagung des GI-Fachbereichs Softwaretechnik

P-214 Volker Markl, Gunter Saake, Kai-Uwe Sattler, Gregor Hackenbroich, Bernhard Mit schang, Theo Härder, Veit Köppen (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 2013
13. – 15. März 2013, Magdeburg

P-215 Stefan Wagner, Horst Lichter (Hrsg.)
Software Engineering 2013
Workshopband
(inkl. Doktorandensymposium)
26. Februar – 1. März 2013, Aachen

P-216 Gunter Saake, Andreas Henrich, Wolfgang Lehner, Thomas Neumann, Veit Köppen (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband
11. – 12. März 2013, Magdeburg

P-217 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
6. DFN-Forum Kommunikations-technologien
Beiträge der Fachtagung
03.–04. Juni 2013, Erlangen

P-218 Andreas Breiter, Christoph Rensing (Hrsg.)
DeLFI 2013: Die 11 e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
8. – 11. September 2013, Bremen

P-219 Norbert Breier, Peer Stechert, Thomas Wilke (Hrsg.)
Informatik erweitert Horizonte
INFOS 2013
15. GI-Fachtagung Informatik und Schule
26. – 28. September 2013

P-220 Matthias Horbach (Hrsg.)
INFORMATIK 2013
Informatik angepasst an Mensch, Organisation und Umwelt
16. – 20. September 2013, Koblenz

P-221 Maria A. Wimmer, Marijn Janssen, Ann Macintosh, Hans Jochen Scholl, Efthimios Tambouris (Eds.)
Electronic Government and Electronic Participation
Joint Proceedings of Ongoing Research of IFIP EGOV and IFIP ePart 2013
16. – 19. September 2013, Koblenz

P-222 Reinhard Jung, Manfred Reichert (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2013)
St. Gallen, Switzerland
September 5. – 6. 2013

P-223 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2013
10. – 11. September 2013
Kloster Banz, Germany

P-224 Eckhart Hanser, Martin Mikusz, Masud Fazal-Baqaie (Hrsg.)
Vorgehensmodelle 2013
Vorgehensmodelle – Anspruch und Wirklichkeit
20. Tagung der Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V.
Lörrach, 2013

P-225 Hans-Georg Fill, Dimitris Karagiannis, Ulrich Reimer (Hrsg.)
Modellierung 2014
19. – 21. März 2014, Wien

P-226 M. Clasen, M. Hamer, S. Lehnert, B. Petersen, B. Theuvsen (Hrsg.)
IT-Standards in der Agrar- und Ernährungswirtschaft Fokus: Risiko- und Krisenmanagement
Referate der 34. GIL-Jahrestagung
24. – 25. Februar 2014, Bonn

P-227 Wilhelm Hasselbring,
Nils Christian Ehmke (Hrsg.)
Software Engineering 2014
Fachtagung des GI-Fachbereichs
Softwaretechnik
25. – 28. Februar 2014
Kiel, Deutschland

P-228 Stefan Katzenbeisser, Volkmar Lotz,
Edgar Weippl (Hrsg.)
Sicherheit 2014
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 7. Jahrestagung des
Fachbereichs Sicherheit der
Gesellschaft für Informatik e.V. (GI)
19. – 21. März 2014, Wien

P-230 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2014
Proceedings of the 13th International
Conference of the Biometrics Special
Interest Group
10. – 12. September 2014 in
Darmstadt, Germany

P-231 Paul Müller, Bernhard Neumair,
Helmut Reiser, Gabi Dreo Rodosek
(Hrsg.)
7. DFN-Forum
Kommunikationstechnologien
16. – 17. Juni 2014
Fulda

P-232 E. Plödereder, L. Grunske, E. Schneider,
D. Ull (Hrsg.)
INFORMATIK 2014
Big Data – Komplexität meistern
22. – 26. September 2014
Stuttgart

P-233 Stephan Trahasch, Rolf Plötzner, Gerhard
Schneider, Claudia Gayer, Daniel Sassiat,
Nicole Wöhrle (Hrsg.)
DeLFI 2014 – Die 12. e-Learning
Fachtagung Informatik
der Gesellschaft für Informatik e.V.
15. – 17. September 2014
Freiburg

P-234 Fernand Feltz, Bela Mutschler, Benoît
Otjacques (Eds.)
Enterprise Modelling and Information
Systems Architectures
(EMISA 2014)
Luxembourg, September 25-26, 2014

P-235 Robert Giegerich,
Ralf Hofestädt,
Tim W. Nattkemper (Eds.)
German Conference on
Bioinformatics 2014
September 28 – October 1
Bielefeld, Germany

P-236 Martin Engstler, Eckhart Hanser,
Martin Mikusz, Georg Herzwurm (Hrsg.)
Projektmanagement und
Vorgehensmodelle 2014
Soziale Aspekte und Standardisierung
Gemeinsame Tagung der Fachgruppen
Projektmanagement (WI-PM) und
Vorgehensmodelle (WI-VM) im
Fachgebiet Wirtschaftsinformatik der
Gesellschaft für Informatik e.V., Stuttgart
2014

P-237 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2014
4.–6. November 2014
Stuttgart, Germany

P-238 Arno Ruckelshausen, Hans-Peter
Schwarz, Brigitte Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und
Ernährungswirtschaft
Referate der 35. GIL-Jahrestagung
23. – 24. Februar 2015, Geisenheim

P-241 Thomas Seidl, Norbert Ritter,
Harald Schöning, Kai-Uwe Sattler,
Theo Härder, Steffen Friedrich,
Wolfram Wingerath (Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2015)
04. – 06. März 2015, Hamburg

P-242 Norbert Ritter, Andreas Henrich,
Wolfgang Lehner, Andreas Thor,
Steffen Friedrich, Wolfram Wingerath
(Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2015) –
Workshopband
02. – 03. März 2015, Hamburg

The titles can be purchased at:

Köllen Druck + Verlag GmbH
Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn
Fax: +49 (0)228/9898222
E-Mail: druckverlag@koellen.de

Gesellschaft für Informatik e.V. (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into
- seminars
- proceedings
- dissertations
- thematics

current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: http://www.gi.de/service/publikationen/lni/

ISSN 1617-5468
ISBN 978-3-88579-636-7

The BTW 2015 in Hamburg is the 16th conference of its kind reflecting the broad range of academic research and industrial development work within the German database community. Some of the various hot topics of this 2015 conference are: new hardware and memory technologies, information extraction, information integration, big data analytics, web data management, service-oriented computing and cloud computing. This volume contains contributions from the refereed workshop program, the refereed student program, and tutorials.