HPI Future SOC Lab: Proceedings 2016

Christoph Meinel, Andreas Polze, Gerhard Oswald,
Rolf Strotmann, Ulrich Seibold, Bernhard Schulzki (Eds.)

HPI Future SOC Lab:
Proceedings 2016

Christoph Meinel | Andreas Polze | Gerhard Oswald | Rolf Strotmann |
Ulrich Seibold | Bernhard Schulzki (Eds.)

HPI Future SOC Lab

Proceedings 2016

Hasso-Plattner-Institut für Softwaresystemtechnik GmbH

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de/ abrufbar.

Hasso-Plattner-Institut 2018
https://hpi.de/

Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam
Tel.: +49-(0)331 5509-0 7 / Fax: +49-(0)331 5509-325
E-Mail: hpi-info@hpi.de

Online veröffentlicht auf dem Publikationsserver der Universität Potsdam
URN urn:nbn:de:kobv:517-opus4-406787
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus4-406787

Contents

The Structure of Industrial SAT Instances

Tobias Friedrich and Andrew M. Sutton

Hasso Plattner Institute, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam
{tobias.friedrich,andrew.sutton}@hpi.de

Abstract

Many computer science problems can be encoded as propositional formula and solved by determining their satisfiability (SAT). Despite negative worst-case complexity results, many large industrial SAT instances can be solved efficiently by modern solvers. The goal of this project was to study the structure and hardness of industrial SAT instances as well as random SAT instances generated from non-uniform distributions. Our goal is to determine what properties are essential for efficient SAT solving.

1 Introduction

Propositional satisfiability (SAT) is one of the most fundamental problems in computer science. Many practical questions from different domains can be encoded as propositional formula and solved by determining the satisfiability of the resulting formula. A propositional formula is constructed from a set V of n Boolean variables by forming a conjunction

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$$

of m disjunctive clauses where

$$C_i = (\ell_1 \vee \ell_2 \vee \cdots \vee \ell_{k_i}).$$

where $\ell_j \in \{v, \neg v\}$ for some $v \in V$. Here $\neg v$ denotes the logical negation of v. The goal of the decision problem is to decide if there is an assignment to all variables of V so that F evaluates to true. SAT is a central problem in theoretical computer science, but it is also an important practical problem since many difficult combinatorial problems reduce to it.

SAT instances and distributions. A distribution of SAT instances is typically parameterized by n and m and is described by a categorical distribution over all formulas over n variables and m clauses. The most heavily studied distribution of SAT instances is the *uniform* distribution. The uniform distribution is the distribution $U_{n,m}$ of all well-formed CNF formulas on n variables and m clauses where each formula has the same probability of being selected.

Most theoretical work on SAT instances has focused almost exclusively on this uniform distribution $U_{n,m}$. Uniform random formulas are easy to construct, and have shown to be accessible to probabilistic analysis due to their statistical uniformity. Indeed, a long line of successful research has relied on the uniform distribution, and from it, several sophisticated rigorous and non-rigorous techniques have developed for analyzing random structures in general.

Nevertheless, a focus on uniform random instances comes with a risk of driving SAT research in the wrong direction [9] because such instances do not possess the same structural properties as ones encountered in practice. It is well-known that solvers that have been tuned to perform well on one class of instances do not necessarily perform well on another [3], and studying the algorithmics of solvers on uniform random formulas can lead research astray.

The empirical SAT community has expanded their view to study *industrial* instances. Industrial instances arise from problems in practice, such as hardware and software verification, automated planning and scheduling, and circuit design. Empirically, industrial instances appear to have strongly different properties than formulas generated uniformly at random, and as might be expected, SAT solvers behave very differently when applied to them [7, 10].

Furthermore, a number of *non-uniform* random distributions have been recently proposed. These models include regular random [4], geometric [5] and scale-free [1, 2]. The scale-free model is especially promising because the *degree distribution* (distribution of variable occurrence) of instances follows a power-law and this phenomenon has been observed on real-world industrial instances.

Project aim. The goal of this project was to utilize the parallel computing power of the 1000 node cluster

of the Future SOC Lab to (1) measure the empirical degree distribution of large industrial instances and (2) generate a massive set of large random non-uniform (scale-free) formulas and run a SAT solver on them to check their satisfiability and hardness.

2 Industrial instances

We measured the empirical degree distribution of 300 instances from the main track of the SAT Race 2015 competition (`http://baldur.iti.kit.edu/sat-race-2015/`). In contrast to other SAT competitions, SAT Race has a focus on application instances instead of synthetic instances. For each formula, we count the occurrence of each variable and calculate the empirical cumulative degree distribution. We observed that the degree distribution of several groups of formulas related to bounded model checking often exhibited a power-law. In Figure 1 we plot two representative groups: `hwmcc10` (hardware model checking) and `SAT_dat` (IBM formal verification suite). We also compared these with the degree distributions of a random scale-free instance (see Section 3) with $n = 10^6$ variables, $m = 4.5 \times 10^6$ clauses, and power law exponent $\beta = 2.75$, as well as a uniform random formula ($n = 10^6$, $m = 4.5 \times 10^6$). The tail of the distributions that appear linear in the double log plot are following a power-law.

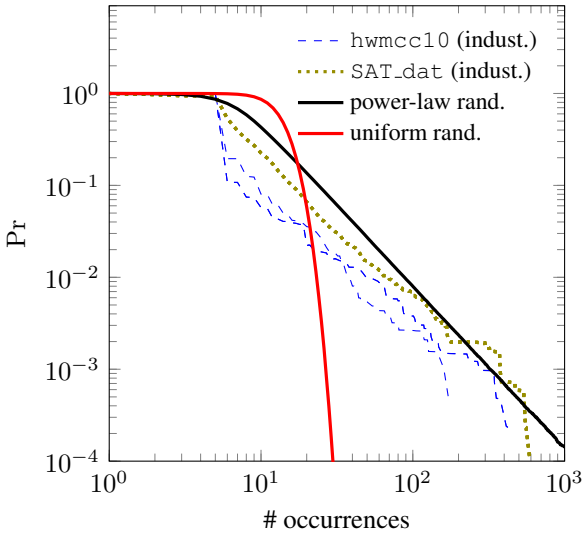


Figure 1: Cumulative variable occurrence distributions of two industrial categories from SAT Race 2015 compared to a random scale-free k-CNF formula ($n = 10^6$ variables, $m = 4.5 \times 10^6$ clauses, power law exponent $\beta = 2.75$) and a uniform random formula ($n = 10^6$, $m = 4.5 \times 10^6$). The parameters for the industrial instances range from $n \approx 8.8 \times 10^4$ to $n \approx 1.5 \times 10^6$ and $m \approx 2.6 \times 10^5$ to $m \approx 6 \times 10^6$. The distributions of the industrial instances are much closer to the random power-law formula than to the uniform random formula.

3 Non-uniform random instances

In this section, we report on results for the *scale-free* instance distribution. In particular, a scale-free formula F on n variables and m clauses of length k is constructed as follows. We define a set of n weights

$$w_i := \left(\frac{n}{i}\right)^{\frac{1}{\beta-1}}, \quad \text{for each } i \in [n],$$

where $\beta > 0$ is a parameter called the *power-law exponent*. Let $V = \{v_1, v_2, \ldots, v_n\}$ denote the set of variables. Rather than picking each variable uniformly at random to construct a clause, we select variable v_i with probability

$$p_i = \frac{w_i}{\sum_{j=1}^{n} w_j}.$$

Each of the m clauses is then sampled independently at random using $\{p_i : i \in [n]\}$ to sample the variables as follows:

1. Select k variables independently at random from the distribution $\{p_i : i \in [n]\}$. Repeat until no variables coincide.
2. Negate each of the k variables independently at random with probability $1/2$.

Thus each such formula is generated at random, but the resulting degree distribution follows a power-law with exponent β.

We used GNU Parallel [11] to distribute a large number of jobs over the cluster. Each job was responsible for generating a set of random scale-free formulas, and then attempting to solve each within some predetermined time limit. In an interest to eliminate statistical fluctuations that sometimes arise at small problem sizes, we set n very large, specifically $n = 10^6$. For each power-law exponent $\beta = 1.5, 1.6, \ldots, 3.5$ and each m such that $m/n = 1/10, \ldots, 10$ we generated 50 scale-free formulas in the above manner, and ran the CDCL SAT solver MiniSAT [8] with a time out of 15 minutes (900 seconds). If the satisfiability of the formula cannot be determined within this time, the formula is marked as "hard", and its satisfiability state is unknown.

Our main goal for this part of the project was to determine the nature of the so-called phase transition from satisfiable to unsatisfiable formulas. Specifically, we are interested in two major phenomena. First, we want to determine the location of the threshold as a function of both constraint density (measured by m/n, or clause to variable ratio) and power-law exponent β. Second, we want to measure where the *hard* instances are. In the uniform case, for example, it has been suggested that hard formulas lie at the phase transition [6]. To observe the location of the phase transition as a function of constraint density and power-law exponent, we report a representative phase diagram on the

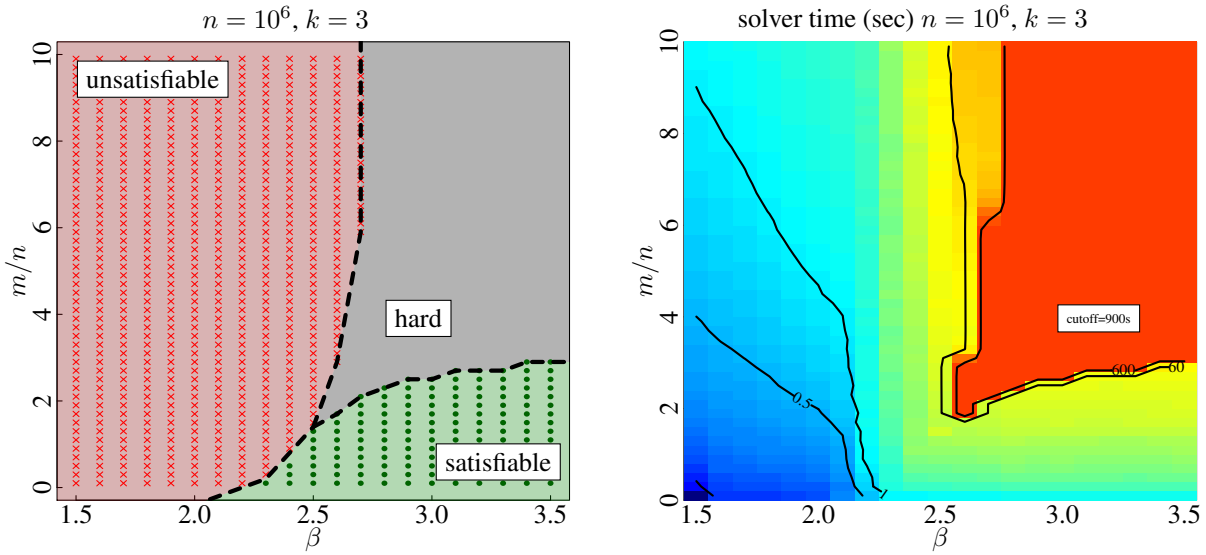


Figure 2: Phase diagram (left) and timing contour plot (right) for scale-free 3-SAT with $n = 10^6$. We see a phase transition from unsatisfiable (×) to satisfiable (●) and a patch of very hard instances (■), close to the phase transition for higher m/n and β. The heat map (right) reports mean solver time on the formulas (blue=fast,red=slow); solver run time strongly increases around the phase transition.

left in Figure 2 for clause length $k = 3$. In this picture, each point corresponds to a set of 50 formulas at a given density and scale parameter. If all 50 formulas were unsatisfiable, a red cross (×) is drawn. If some formulas are satisfiable, a green dot (●) is drawn with the size of the dot corresponding to the fraction of satisfiable instances. Note that the threshold appears to be sharp, and in most cases, either all formulas were satisfiable, all were unsatisfiable, or all were hard.
To understand the location of hard formulas, we report the timing data as a contour plot of the mean solver time (in seconds) required by MiniSAT as a function of the $(\beta, m/n)$-plane.

4 Conclusions and outlook

With this project we were able to assess the shape and location of the phase transition in a massive set of random formulas generated by a non-uniform distribution. We identified the troubling spots of the distribution in terms of constraint density and power-law exponent.
Regarding the technical requirements: Scheduling jobs on the FSOC cluster turned out to be a non-trivial task. We look forward for the new batch scheduler implemented in the next round of the Future SOC lab.

References

[1] C. Ansótegui, M. L. Bonet, and J. Levy. On the structure of industrial SAT instances. In *15th CP*, pp. 127–141, 2009.

[2] C. Ansótegui, M. L. Bonet, and J. Levy. Towards industrial-like random SAT instances. In *21st IJCAI*, pp. 387–392, 2009.

[3] M. Birattari. *Tuning Metaheuristics: A Machine Learning Perspective*. Springer, Berlin Heidelberg, 2009.

[4] Y. Boufkhad, O. Dubois, Y. Interian, and B. Selman. Regular random k-SAT: Properties of balanced formulas. In *9th SAT*, pp. 181–200, 2006.

[5] M. Bradonjic and W. Perkins. On sharp thresholds in random geometric graphs. In *18th Intl. Workshop on Randomization and Computation (RANDOM)*, pp. 500–514, 2014.

[6] J. M. Crawford and L. D. Auton. Experimental results on the crossover point in random 3-SAT. *Artificial Intelligence*, 81:31 – 57, 1996.

[7] J. M. Crawford and A. B. Baker. Experimental results on the application of satisfiability algorithms to scheduling problems. In *12th AAAI*, pp. 1092–1097, 1994.

[8] N. Eén and N. Sörensson. An extensible SAT-solver. In *7th SAT*, pp. 502–518, 2004.

[9] H. Kautz and B. Selman. The state of SAT. *Disc. Appl. Math.*, 155:1514–1524, 2007.

[10] K. Konolige. Easy to be hard: Difficult problems for greedy algorithms. In *4th KR*, pp. 374–378, 1994.

[11] O. Tange. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36: 42–47, 2011.

Sentiment Analysis on Twitter Data Using R Algorithms

Marlene Knigge
Technical University of Munich
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
marlene.knigge@in.tum.de

Christopher Kohl
Technical University of Munich
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
christopher.kohl@in.tum.de

Galina Baader
Technical University of Munich
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
galina.baader@in.tum.de

Harald Kienegger
Technical University of Munich
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
harald.kienegger@in.tum.de

Helmut Krcmar
Technical University of Munich
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
krcmar@in.tum.de

Abstract

The goal of our project is to identify perceived risks and emotions of autonomous driving based on Twitter data (i.e. Tweets). While autonomous driving is not only a vision anymore but becoming part of our daily live, pros and cons are discussed controversially. Fears and resistance to autonomous driving may be justified in some areas, in others they emerge without sound reasons. For stakeholders of autonomous driving, knowing people's opinion is valuable as they can react to unfounded fears and try to prevent resistance to autonomous driving. In order to provide them with suitable data, we extracted Tweets from Twitter and applied different text mining algorithms to them. While we were able to gain useful results using the native SAP HANA and R algorithms, we are still facing issues with three PAL algorithms. Therefore, we apply for a follow-up project.[1]

1 Introduction

Today, people from all over the world are able to interact via social networks – e. g. Twitter. They exchange unstructured information including texts and pictures concerning a huge variety of topics. Companies have discovered that social media data may be valuable for them if they either achieve to extract information relevant to their business or use social media to transmit their news to potential customers [1]. An inconceivable amount (volume) of unstructured (variety) data thus is generated and stored every day (velocity) [2]. Therefore, social media data can be considered as big data. Hardware, applications, and algorithms are evolving which allows the analysis of these huge amounts of unstructured data, e. g. in-memory databases and specialized machine learning algorithms that can be used for text mining.

In most industries, the digital transformation is imminent. Thus, knowledge about public sentiments towards digitized products, such as autonomous driving cars, is invaluable for industry as well as for research. While some people are excited by the idea of autonomous driving, others feel concerned or even afraid about it. It has been shown in different fields that concerns and anxiety may lead to resistance to the use of new technologies, which companies may

[1] For further information, please read our application for the follow-up project at HPI future SOC lab.

want to prevent [3]. Therefore, they may use information extracted from social networks as this allows them real-time access to opinions of potential customers. Gathering this information from traditional surveys takes a lot of time and suffers from low response rates [4, 5]. Manually observing and evaluating all social media conversations is almost impossible due to the enormous and still increasing amounts of data generated each day [6].

2 Project Goal

We aim to contribute to the analysis of unstructured data, especially data originating from social networks and, therefore, improve the application of text mining algorithms.

The extraction of data from social networks as well as the analysis of the data with text mining tools is an emerging field. We applied different native SAP HANA, SAP HANA Predictive Analysis Library (PAL) and R algorithms to a data sample comprising data from Twitter concerning autonomous driving. We then compared and evaluated the results of each algorithm we applied.

3 Project Design

Our project comprised the steps described in Figure 1, which we performed for each of the algorithms we examined.

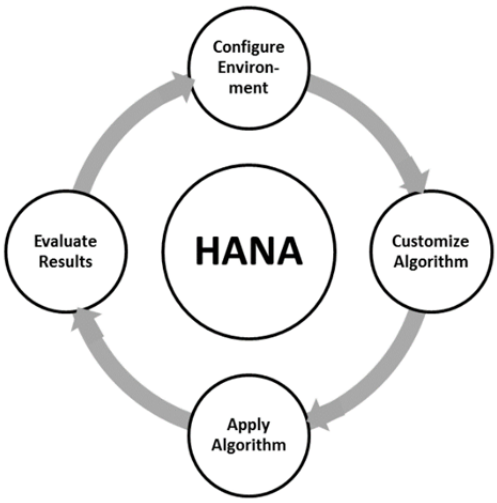


Figure 1: Project steps (Source: Own illustration)

First, we conducted some preliminary preparing steps like the generation of a text mining index, embedding the PAL, installing R libraries, and adding authorization roles needed for applying the algorithms. Second, we customized the algorithm by setting the parameters, e. g. the number of considered neighbors for the K Nearest Neighbor algorithm or the kernel of the Support Vector Machine algorithm. Third, we applied the algorithm to the training data (cp. section 3.1). Fourth, we evaluated the results, mainly by, for example, analyzing how many Tweets of the training data have been categorized correctly by the algorithm. In further iterations, we changed the parameters and applied the trained algorithms to the main data sample.

3.1 System configuration

We were provided with a SAP HANA SPS10 with 1 TB RAM and 32 Cores (CPU). Additionally, we were provided with the SAP HANA Predictive Analysis Library (PAL) and access to an R Server. Our main tool for applying algorithms was the SAP HANA Studio, Version 2.0.11 which is Eclipse-based.

3.2 Underlying Datasets

We used two datasets which we collected in previous work on which we applied text mining algorithms: a training data set for evaluating and configuring the algorithms and a much more extensive dataset for getting substantive results.

The training data was provided in a csv-file comprising 7,500 Tweets about autonomous driving, which have been classified manually before so that they can be used to train the algorithms. They have been assigned to the classes “benefit” (750 Tweets), “neutral” (6000 Tweets), and “risk” (750 Tweets). When applying the algorithm on these, the quality of the results can be judged by comparing the actual classification of the Tweets with the results of the algorithms applied. More precisely, a (random) subset of the training data is used for training the classifier, while the remaining data is used for the evaluation. While importing the training data and the main dataset into tables in SAP HANA, we were facing several issues. Therefore, we had to modify the structure of the files to be able to import the data.

The first problem was that the csv-file was comma-separated, which led to an error if a Tweet contained a comma. It was not possible to specify in the import dialogue of SAP HANA Studio that strings are enclosed in parentheses so that commas in the Tweets would not be parsed as separators. Therefore, we changed the separator of the csv-file from comma to the tabulator character. If a Tweet contained a tabulator character, it was replaced with a simple space character. We chose the tabulator character as we did not expect that to change the meaning of the Tweet. This is important since Text Analysis of SAP HANA does not treat texts as a simple bag of words but uses a more sophisticated analysis of the sentence structure. The second problem occurred when we tried to import the main dataset with SAP HANA Studio: The application crashed while uploading the data. Therefore, we implemented a Java application that uses the JDBC connector of the SAP HANA database to insert the Tweets.

3.3 Applied Algorithms and Results

In total, we examined six algorithms so far as shown in Figure 2. We examined two native SAP HANA algorithms: Text Mining based on the K-Nearest Neighbor (KNN) and Text Analysis using Voice of Customer (VoC). Furthermore, we examined three PAL algorithms: Naive Bayes, Support Vector Machine (SVM), and C4.5 Decision Tree. At last, we examined one R algorithm for SVM.

For each algorithm, we first used the training data to evaluate the performance of the classifier. As this evaluation was our main focus we will only refer to the results using the training dataset in the following. For an overview of the results please see Table 1.

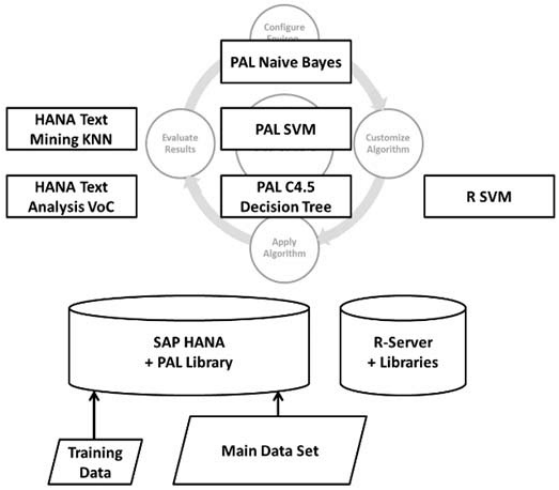


Figure 2: Test Environment (Source: Own illustration)

Some Tweets cannot be classified by Text Mining (KNN). When we left them out, Text Mining (KNN) classified 90 % of the Tweets correctly. If we included them, Text Mining (KNN) still classified about 85 % correctly.

Unlike the other algorithms, Text Analysis (VoC) does not assign the Tweets to the classes "benefit", "neutral", and "risk" but it uses the following classes: "MinorProblem", "MajorProblem", "WeakNegativeSentiment", "WeakPositiveSentiment", "StrongNegativeSentiment" and "StrongPositiveSentiment". So the results are not directly comparable with those of the other algorithms, but we can tell that they are still useful for extracting meaning from the Tweets.

The PAL algorithm C4.5 Decision Tree is not useful for analyzing Tweets, as it generates a branch out of each Tweet so that we do not get a meaningful decision tree as result. This is due to the fact that it expects several columns for input, while Tweets are only stored in one column.

Applying the PAL algorithms Naive Bayes and SVM leads to similar results. We tested both with a subset of the training data: Out of about 250 Tweets which have been pre-classified as "risk", only one has been found. Out of about 250 Tweets assigned to the class "benefit", only 15 have been identified. We noticed, that if the training dataset is not well-balanced, i.e. it does not contain 1/3 of each, "risk", "neutral" and "benefit"-classified Tweets, it will tend to assign Tweets to the group that has been overrepresented in the training data. This has to be taken into account when working with these algorithms. It may lead to erroneous results. So we conclude that the quality of the analysis using these algorithms is not quite precise. We believe that this is due to the fact that we could not use the text mining index, which is a Document-Term-Matrix, to train these algorithms.

With SAP HANA SPS11, further algorithms will be available like Random Forests, Area-Under-Curve, and Predict With Tree Model. As we are currently working on SAP HANA SPS10, we have not been able to examine those yet.

Using R, a Document-Term-Matrix can be created on the R Server when using the package RTextTools. Unfortunately, the Document-Term-Matrix can only be used inside R since we have not been able to transfer the matrix back to the HANA database. The problem is that we do not know the dimensions of the Document-Term-Matrix, which will be created by R, in advance. Although, we still are facing technical issues when applying the algorithm to the data, our first impression is that the classification using SVM in R leads to acceptable results with more than 72 % of correct classifications on a subset of the training data.

Table 1: Overview of algorithms applied and results (Source: Own presentation)

Algorithms for Analyzing Tweets

	Algorithm	Applic-	Issues	Findings	Outlook/Next Steps
Native HANA	Text Mining (KNN)	X	- Implicit text mining index created by HANA can only be used inside this algorithm	85 - 90% of the Tweets classified correctly - Best of all evaluated algorithms for text classification so far	- Create a customized text mining and text analysis configuration file for improved extraction of information from Tweets
	Text Analysis (Voice of Customer configuration)	X	- Not comparable to other algorithms but it provides meaningful results as well	- Very good for getting a first impression of the overall sentiments	- Derive sentiment of Tweets from the sentiments of the included terms - Analyze correlations between the sentiments and the classes "risk", "benefit", and "neutral"
PAL	Naive Bayes	X	- No document-term-matrix is generated inside these algorithms. The Tweet-string is used for classification.	- < 1% of "risk" and only 6 % of "benefit" classified correctly - Results strongly depend on the distribution of classes in the training set. Best results with a uniform distribution of classes in the training set	- Generate an explicit document-term-matrix with HANA or R - Apply the PAL-algorithms to the document-term-matrix
	SVM	X			
	C4.5 Decision Tree	O	- Not applicable for analyzing Tweets as it generates a branch out of each Tweet. Therefore, no meaningful decision tree is generated		
	Random Forests, Area-Under-Curve, Predict With Tree Model	-	- Not available in HANA SPS10.	- Not available in HANA SPS10.	- Update HANA from SPS10 to SPS11 - (Re-)Evaluate the existing and the new algorithms
R	SVM	X	- All Tweets need to be transfered as strings to the R server - The document-term-matrix generated with R can only be used inside R and cannot be transfered back to the HANA, e.g. for reuse	'- Approximately 83 % of Tweets classified correctly	- Transfer and use text mining index from HANA in R - Transfer and use document-term-matrix from R in HANA - Optimization of the parameters of the R algorithms
	Decision Tree	X		- Approximately 60 % of Tweets classified correctly	

4 Conclusion and Outlook

While we had collected our datasets before starting the project, we were facing a lot of issues before we have been able to apply the algorithms on our data, such as missing or incomplete add-ons, missing authorizations for using the features offered by the add-ons or issues when trying to upload the data without preprocessing. Unfortunately, the error messages did not always make clear what the cause of an issue was, so we spend a lot of time in trying out and finding out how to configure our system and to prepare our datasets.

From our first sampling we can tell that using SAP HANA native Text Mining (KNN) led to the best results. The results using Text Analysis (VoC) of native SAP HANA led to useful results as well but it is difficult to compare these results with the classification approach. The PAL algorithms Naive Bayes, SVM, and C4.5 Decision Tree did not lead to acceptable results. The classification with the SVM on the R server led to good results, however, we were only able to test the application very shortly and identified minor issues, which we have not been able to fix so far.

Our next steps will be to proceed with and refine the application of text mining algorithms, especially going more into detail using the R algorithms on the one hand. Furthermore, we plan to apply the algorithms, which provide reasonable results, to the main dataset. On the other hand, we aim to extend our analysis by examining the possibilities provided by entity and facts extraction included SAP HANA text analysis.

References

[1] *Gabler Wirtschaftslexikon, Stichwort: Soziale Medien*. [cited 2015 15.09.]; Available from: http://wirtschaftslexikon.gabler.de/Archiv/569839/soziale-medien-v2.html.

[2] Russom, P., *Big Data Analytics*. TDWI Best Practices Report, Fourth Quarter, 2011: p. 1-35.

[3] Sanford, C. and H. Oh, *The Role of User Resistance in the Adoption of a Mobile Data Service*. CyberPsychology, Behavior & Social Networking, 2010. **13**(6): p. 663-672.

[4] Klie, L., *Listening to the Voice of the Customer*. CRM Magazine, 2012. **16**(1).

[5] Musico, C., *The Feedback Funnel.* CRM Magazine, 2009. **13**(1): p. 27-31.
[6] Snijders, C., U. Matzat, and U.-D. Reips, *Big Data: Big Gaps of Knowledge in the Field of Internet Science.* International Journal of Internet Science, 2012. **7**(1): p. 1-5.

ActOnAir
Data Mining of Environmental Data and Bio-signals

Matthias Scholz
Hochschule Mainz
Lucy-Hillebrand-Straße 2
55128 Mainz
matthias.scholz@hs-mainz.de

Gunther Piller
Hochschule Mainz
Lucy-Hillebrand-Straße 2
55128 Mainz
gunther.piller@hs-mainz.de

Abstract

Goal of the project ActOnAir is the personal guidance of individuals who suffer from asthma and need to reduce their exposure to air pollutants. For this purpose bio-signals and environmental data are captured and analyzed with different data mining techniques. Resulting classification models can then be used for real-time predictions of health risks. This contribution describes the data mining components which are implemented on the in-memory platform SAP HANA.

1 Introduction

The project ActOnAir has already been introduced in a previous report [1]. This earlier paper described the research question and the architecture of the overall IT system. A brief summary of these topics is added in this publication for completeness.

Focus of ActOnAir is a personal guidance of asthma patients to reduce their risk for asthma attacks. The design objectives of the overall hard- and software system are:

- The burden of individuals suffering from asthma shall be captured in a comprehensive and detailed way. The personal exposure of patients to air pollutants and environmental factors is measured in short time intervals and correlated to up-to-date individual health factors.
- Personal guidance to patients shall be provided in real-time. It is tailored to the constitution and the current situation of the individual.

These objectives were starting point for the identification of detailed requirements for the IT system and the subsequent design of its overall architecture as described in [1, 2]. In summary, the ActOnAir system consists of five major components:

- Mobile Sensor Box for the capture and transmission of the individual exposure of persons to air pollutants
- Sensor Integration and Geo Sensor Network for the processing and storage of heterogeneous sensor data
- Data Mining and Forecasting for the identification of frequent sequential patterns for health-related factors and the derivation of forecast models
- Mobile Application for the recording of personal health symptoms and the provisioning of real-time forecasts for health risks
- Mobile Cloud Computing Services for the provisioning of common services, like user management and authorization

In the following we focus on the component for data mining and forecasting. First, the data mining approach is introduced. Then its implementation upon the SAP HANA platform is described.

2 Data Mining Method

As a starting point we follow the Pattern Based Decision Tree method of Lee et al. [3]. Here sequential pattern mining is integrated with classification through decision tree mining. Figure 1 illustrates this approach.

First, patient information and health data are combined with environmental measurements into patient datasets. These are then prepared for data mining. Important steps are a possible segmentation of patient data into buckets with similar health characteristics and the discretization of sensor measurements. For example, the values for relative humidity could be split into four different bins.

As a next step sequential pattern mining is used to identify frequent sequences of environmental factors

and bio-signals before moments with and without asthma attacks.

These frequent sequential patterns are then treated as features to characterize the input patient datasets. From these feature sets decision trees are derived for different patient segments or individuals.

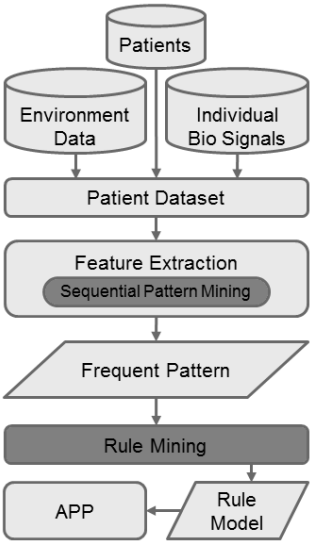


Figure 1: Data Mining Method

Finally, the decision trees serve as a set of rules provisioned to the smartphone applications of end-users. With their help actual measurements of environmental factors and individual health symptoms can be evaluated in real-time. Situations with high risk for forthcoming asthma attacks can then be directly identified.

3 Sequential Pattern Mining

In the analyses for ActOnAir we consider sequences $S = \langle s_1, s_2, \dots s_n \rangle$ of temporally ordered values for environmental factors and individual bio-signals.[1] As a starting point one entry per day is used for all considered factors. For example, a sequence of length three for temperature contains the discretized temperature values for three subsequent days, e.g. $\langle cold, medium, high \rangle$. Depending on the nature of a particular influence factor, the chosen daily entry can be determined differently, e.g. as an average, a minimum or maximum, or through accumulation.

For a specific patient and a particular influencing factor all sequences of length *n*, e.g. *n = 3 days*, before days with an asthma attack are collected within a dataset for high-risk sequences D_h. Sequences of similar length before days without attacks build a dataset D_l of low-risk sequences. This selection and distribution of sequences is carried out for all considered environmental factors and bio-signals. Sequential pattern mining is then performed for all factors separately within high- and low-risk segments, i.e. based on the corresponding datasets D_h and D_l, respectively.

In our approach we search for frequent sequential patterns with length from one to *n*. For sequences with length $1 < m \leq n$ we also consider patterns containing at most $m - 1$ arbitrary entries. In this way it can be found out, whether certain health factors are effective over a period of several days. For example, an extraordinary exposure to particulate matter on a certain day is likely to increases the risk for an asthma attack for several forthcoming days – even if days after the high exposure are spent in an environment with a low concentration of particulate matter.

A common measure for the significance of a frequent sequential pattern *S* is its occurrence or support, denoted by $\sigma\,(S, D)$. It amounts to the total number of input-sequences in the database *D* that contain *S* as a subsequence (see e.g. [4]). Important for the relevance of a sequence *S* are the measures *confidence* and *lift*. In our case *confidence* describes the conditional probability that a sequence within the complete dataset $D = D_h + D_l$, which contains *S* as a subsequence, is found within the set of high-risk sequences D_h. Expressed through the support of *S* this means [4, 5]:

$$Conf\,(S) = \frac{\sigma\,(S, D_h)}{\sigma\,(S, D_h + D_l)}$$

Lift measures to what extent the occurrence of sequences with *S* as subsequence is independent from the occurrence of sequences within the high-risk dataset D_h, i.e. [5]:

$$Lift\,(S) = \frac{\sigma\,(S, D_h)}{\sigma\,(S, D_h + D_l)} \frac{|D_h + D_l|}{|D_h|}$$

Here $|D_{h/l}|$ denotes the total number of input-sequences in D_h and D_l, respectively. Based on the measures *support*, *confidence* and *lift* a meaningful selection of most relevant frequent sequential patterns within the high-risk and low-risk datasets can be carried out. Further ways of efficient pattern pruning can be found in [4, 5, 6].

4 Decision Tree Mining

Following Lee et al. [3], the selected frequent sequential patterns for all factors are interpreted as features. The measured sequences before days with and without asthma attacks are considered as transactions. These are then characterized by the presence or absence of the identified features – which can be interpreted as attributes of the transactions. In addition, with one specific attribute it is described whether an

[1] Only sequences with events made out of single items are considered here.

input-sequence belongs to a high- or low-risk datasets, respectively.

The transactions and their corresponding attributes are then taken as input for decision tree mining. The attribute *high-risk* is set as a target for the mining process. The resulting tree itself is a binary tree. Each node queries the presence of a frequent sequential pattern. A schematic example is shown in Figure 2.

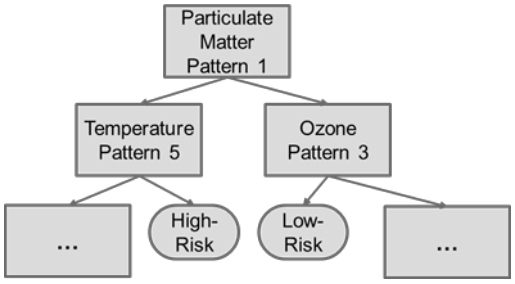


Figure 2: Decision Tree

The decision trees are then used for forecasting. They are provided to the smartphones of patients. Here they are used for the real-time evaluation of actual measurements of environmental factors and biosignals. As a result, a patient knows whether she is in a state of high or low risk for forthcoming asthma attacks.

5 Implementation upon SAP HANA

The module Data Mining and Forecasting is built upon the in-memory platform SAP HANA. Figure 3 shows a sketch of the underlying architecture. The performance of SAP HANA is beneficial for the expected high data volumes: If 5% of asthma patients in Germany would use the application and collect sensor data four times an hour, the approximate data volume would sum up to about 10 GB per day or several TB per year.

Currently the mining process is carried out asynchronously, i.e. independent from the real-time evaluation of health risks. Nevertheless short response times are needed for initial explorative analyses to identify optimal mining parameters, like binning and segmentation, as well as for systematic improvements of forecasting models.

A light weight application for overall data processing and interactive mining steps, e.g. data binning and pattern pruning, has been implemented with SAP HANA Extended Application Services and SAP UI5. Data intensive calculations and data querying are handled through appropriate interfaces in the database using the SQL engine and the Application Function Library with the Predictive Analytics Library (PAL) [7]. As general guideline de-normalized data models have been chosen; write operations were avoided; data intensive application logic has been largely embedded into the database; stored procedures have been parallelized wherever applicable.

Examples for virtual tables are indicated in Figure 3 for person segments, attribute bins and attribute tables. Also several opportunities for parallelization exist: The functional components for binning, segmentation, data pre-processing, pattern identification and sequential pattern mining can be executed independently in parallel for different sensor types and individual patients.

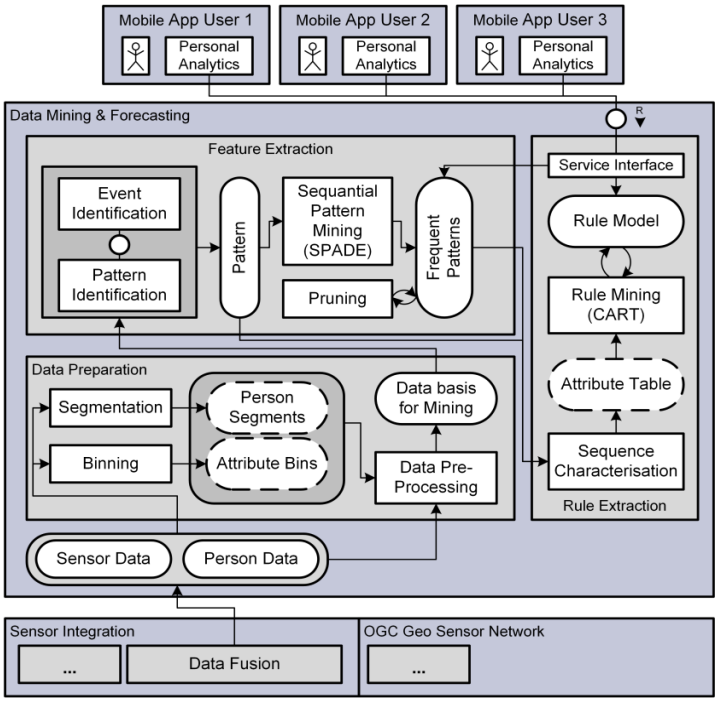


Figure 3: Architecture Data Mining and Forecasting

Data preparation as well as the identification of input-sequences before days with and without asthma attacks is carried out upon HANA itself. For sequential pattern mining an R implementation of the SPADE algorithm is applied [8]. While the R code is embedded in form of a corresponding RLANG procedure, the execution is carried out upon an external R environment. The input-sequences are passed to RLANG procedures through virtual tables. These are transformed into appropriate R data frames. Those data frames are then read into corresponding transaction objects [9]. They serve as final input for the cSPADE algorithm [8]. As input parameters we use in particular [8]:

support	0
maxsize	1
maxlen	3-5
maxgap	1
maxwin	1

Table 1: Parameters cSPADE

The chosen values reflect the consideration of asthma related factors within a window of 3-5 days before asthma attacks and one attribute value per day. Patterns with arbitrary entries for specific days are calculated through appropriate summations. After identified frequent sequential patterns have been obtained, pattern pruning is currently carried out interactively. Here the measures *support*, *confidence* and *lift* are used as key criteria.

For decision tree mining the CART algorithm of PAL [7] has been used. As outlined in Section 4, a table with transactions and corresponding attribute values is used as input. It is obtained by comparing frequent sequential patterns with input-sequences. This can be done efficiently by using the SQL LIKE operator. Target column for the CART algorithm is the attribute value for an asthma attack. It describes whether an input sequence belongs to a high- or low-risk dataset. As output the algorithm provides a table containing the PMML tree model. The model is finally transmitted via an OData interface to the mobile application of an end-user.

6 Status & Next Steps

In March 2016 the status of the overall ActOnAir system is as follows: The data communication platform of the mobile sensor box is available. It can now be tested with various sensors. The sensor for particulate matter is still in development. An asthma diary – which is the main component of the mobile end-user application – is completed for iOS. The component Sensor Integration and Geo Sensor Network provides all basic services for data integration and storage. It has been successfully tested by processing weather data and measurements of air pollutants from publicly available services.

All essential parts of the component for data mining and forecasting have been implemented. This includes functional components for binning, data preprocessing, event and pattern identification, sequential pattern mining, sequence classification and decision tree mining. Functional and performance tests have been carried out with generated test data containing about 2 million datasets. Integration testing of the components Data Mining and Forecasting and Sensor Integration and Geo Sensor Network has been successfully accomplished for migraine data [10]. Also the handover of decision trees to mobile applications in the form of PMML documents has been verified.

The most important next step is a test with realistic data from asthma patients and environmental measuring stations. For this purpose the mobile asthma app is planned to be distributed to test persons, starting end of April 2016. For data mining and forecasting machine learning concepts for an automated optimization of the data mining approach will be investigated. The complete system is planned to be available in Q3 2016.

Supported by the Federal Ministry for Economic Affairs and Energy

References

[1] Scholz M., Bock N., Piller G., Böhm K.: ActOnAir: Data Mining and Forecasting for the Personal Guidance of Asthma Patients. In: Proceedings HPI Future SOC Lab Day Fall 2015. HPI Future SOC Lab, Potsdam, Germany, Universitätsverlag Potsdam, 2016

[2] Bock N., Scholz M., Piller G., Böhm K., Müller H., Fenchel D., Sehlinger T., van Wickeren M., Wiegers W.: Systemarchitektur eines mobilen Empfehlungssystems mit Echtzeitanalysen von Sensordaten für Asthmatiker. In: Proceedings MKWI 2016 Research-in-Progress: 23-29, 2016

[3] Lee C. H. et al.: A Novel Data Mining Mechanism Considering Bio-Signal and Environmental Data with Applications on Asthma Monitoring. Computer Methods and Programs in Biomedicine, 101 (1): 44-61, 2011

[4] Zaki M. J.: SPADE: An efficient algorithm for mining frequent sequences. Machine learning 42 (1-2): 31-60, 2001

[5] Han J., Kamber M., Pei J.: Data Mining: Concepts and Techniques. Elsevier, 2011

[6] Aggrawal, C. C., Han J.: Frequent Pattern Mining. Springer, 2014

[7] SAP PAL: SAP HANA Predictive Analysis Library. http://help.sap.com/hana/sap_hana_predictive_analysis_library_pal_en.pdf, 2015. Last accessed 11th March 2016

[8] Buchta C., Hahsler M., Diaz D.: Package ‘arulesSequences’. https://cran.r-project.org/web/packages/arulesSequences/arulesSequences.pdf, 2015. Last accessed 17th September 2015

[9] Hahsler M.: Package ‘arules’. https://cran.r-project.org/web/packages/arules/arules.pdf, 2015. Last accessed 17th September 2015

[10] Migräne Radar: http://www.migraene-radar.de, 2016. Last accessed 11th March 2016

Natural Language Processing for In-Memory Databases: Multilingual Biomedical Resources

Mariana Neves
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
mariana.neves@hpi.de

Abstract

The biomedical scientific literature is a rich source of information not only in the English language, for which it is more abundant, but also in other languages, such as Portuguese, Spanish and French. We created the first freely available parallel corpus of scientific publications for the biomedical domain. Documents from the Biological Sciences and Health Sciences categories were retrieved from the Scielo database and parallel titles and abstracts are available for the following language pairs: Portuguese/English (about 86,000 documents in total), Spanish/English (about 95,000 documents) and French/English (about 2,000 documents). Additionally, monolingual data was also collected for all four languages. Sentences in the parallel corpus were automatically processed using the HANA database given its support for text analysis for various languages. The corpora are currently being used in the biomedical task in the First Conference on Machine Translation (WMT16).

1 Introduction

Access to the biomedical literature is available on-line via systems such as PubMed®[1], that allow researchers to browse and search for publications of their interest. But articles published in local research journals are accessible only for researchers fluent in the original language of the article, for instance, articles available in databases such as Scielo[2], which has a focus on Latin American publications.

Machine translation (MT) can provide a solution to increase the access to the biomedical literature [9] and to health information in general [4]. Although there has been much work in this field [1], the automatic translation of scientific publications has not received much attention of the community, in part because of the difficulty of getting parallel collections of documents. PubMed is the largest database for scientific publications in biomedicine, however, only titles are available in more than one language in PubMed [10].

Me and some colleagues (Dr. Aurélie Névéol (LIMSI-CNRS, France), Dr. Antonio Jimeno (IBM research, Australia)) created the first freely available parallel collection of scientific publications for the biomedical domain [8]. The documents were derived from Scielo, a database of open access scientific publications with a focus on developing and emerging countries, and especially on Latin America. Scielo currently includes publications in a variety of domains, such as agriculture, engineering, biological and health sciences. It includes abstracts and full texts for publications, mainly in Portuguese and Spanish, but also in English, French and German.

The intended purpose of this parallel corpus is to train and evaluate MT systems. To this end, we created parallel corpora for three pairs of languages: Spanish-English (ES/EN), French-English (FR/EN) and Portuguese-English (PT/EN). These collections are used as training data for the biomedical shared task[3] in the First Conference on Machine Translation (WMT16). Examples of the sentences for all language pairs, i.e., ES/EN, FR/EN and PT/EN, are shown below:

La especie más frecuente aislada de pacientes de ambas regiones fue L. paracasei ssp paracasei 1.
Lactobacillus paracasei ssp paracasei 1 was the most frequently isolated species in both regions.

Le seul traitement validé pour soigner cet état est l'immunothérapie passive avec des sérums antivenimeux d'origine animale sûrs et efficaces.
The only validated treatment for this condition is passive immunotherapy with safe and effective animal-derived antivenoms.

[1] http://www.ncbi.nlm.nih.gov/pubmed
[2] http://www.scielo.org/
[3] http://www.statmt.org/wmt16/biomedical-translation-task.html

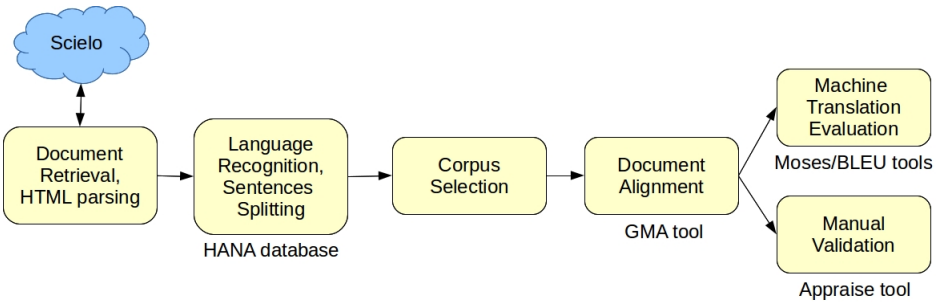


Figure 1: Work-flow of the construction of the parallel collection of biomedical publications.

Avaliação da força muscular periférica de pacientes submetidos à cirurgia cardíaca eletiva: estudo longitudinal.
Evaluation of peripheral muscle strength of patients undergoing elective cardiac surgery: a longitudinal study.

2 Methods

The work-flow for the creation of the parallel corpus is illustrated in Figure 1 and its phases are summarized below.

We developed a crawler for the Scielo web site and retrieved articles periodically from Scielo. We downloaded the page of each article and parsed the HTML code in order to extract the title and the abstract of each publication. Titles and abstracts were subsequently stored and indexed in the SAP HANA database. All publications are available under either the Creative Commons Attribution-Noncommercial 3.0 Unported (cc-by-nc) or Attribution 3.0 Unported (cc-by) licenses, which makes all documents suitable for redistribution and research purposes.

We used the HANA database to perform language recognition in the texts and their segmentation into sentences. Although the language of the publication is usually identified in the Scielo URL, we noticed that there are many situations in which the abstract is in one language and the title in another, making the language recognition step necessary. For instance, the document S0874-48902010000300006[4] contains the abstracts available in French, Spanish and English, four different HTML pages, but the title is always in Portuguese in all of them. The sentence splitting provided by HANA compared favorably to the OpenNLP library[5] on a sample of documents. Further, as stated above, HANA could also be used for language recognition and provides support of various languages, including the ones we focus on in this work.

For both "Biological Sciences" and "Health Sciences" categories, we retrieved from the database pairs of titles and abstracts available in both English and one of the other three languages we consider, i.e., French, Portuguese or Spanish. These constitute the whole collection of parallel documents, which was subsequently split in training and test datasets. Scielo contains many entries only available in one of the languages or in languages other than English, e.g., in both Portuguese and Spanish, given that the focus of the database is in the Latin American journals. These documents constitute our monolingual corpus, given that in-domain monolingual corpora are also a valuable resource for training and evaluation of language models, one of the components of statistical MT systems [5].

We automatically aligned sentences from titles and abstracts for the language pairs using the Geometric Mapping and Alignment (GMA) tool[6]. Me and my colleagues manually checked the automatic alignment generated by the GMA tool to ensure the quality of the corpora. Statistical MT tools need to rely not only on parallel collections of documents, but also on parallel collections of aligned sentences. We randomly selected 100 publications (titles and abstracts) for each category, i.e., Biological Sciences and Health Sciences, and for each of the three pairs of languages, i.e., PT/EN, ES/EN and FR/EN, and manually validated them using the Appraise tool[7]. One of my colleagues (Antonio Jimeno Yepes) trained a statistical MT system using Moses[8][6] on the parallel corpora to demonstrate the capabilities of the proposed corpus.

3 Results

The corpus is currently available for download[9] in the BioC XML format [2], a format which is becoming a standard in the biomedical natural language processing (BioNLP) community. Using this format also ensures the integration of our corpus with tools and other corpora as well as making use of any of the available

[4] http://www.scielo.mec.pt/scielo.php?script=sci_abstract&pid=S0874-48902010000300006&lng=pt&nrm=iso&tlng=pt

[5] https://opennlp.apache.org/

[6] http://nlp.cs.nyu.edu/GMA/

[7] https://github.com/cfedermann/Appraise

[8] http://www.statmt.org/moses

[9] https://drive.google.com/folderview?id=0B3UxRWA52hBja0t2azlkN3d2elk&usp=drive_web

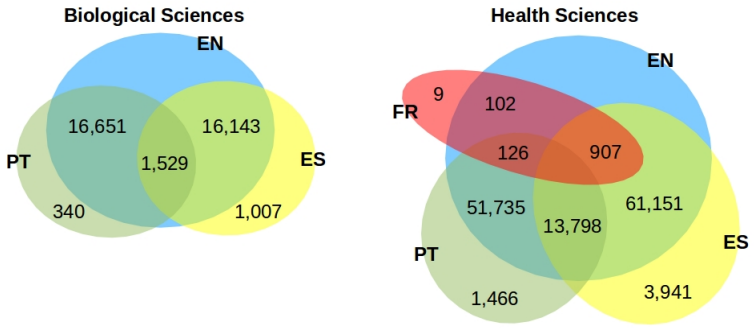


Figure 2: Size of the training datasets for each category, language pair, across language pairs and monolingual corpora. (The figure is not in scale.)

BioC implementations (e.g. Java, Python or C++). The documents are split by sentences according to the analysis we obtained using the HANA database.

As discussed above, the training data is currently being used in the scope of the biomedical task in the WMT16 challenge. Besides the training data, we also released a parallel corpus of MEDLINE titles, similar to the dataset used in our previous work [3], the monolingual documents obtained from Scielo and all alignment output on the sentence and word level that we obtained from the GMA tool.

Due to the focus of the Scielo database on journals from Latin America, the number of documents is much larger for Portuguese and Spanish compared to French, for both categories and for both the parallel and monoligual datasets. Indeed, the number of parallel documents for FR/EN and the Biological Sciences category was so low that we do no provide any training and test datasets for it. Alternatively, it is possible to train a MT system for the Biological Sciences using documents from the Health Sciences, or even completely ignore categories and use a single system trained on the whole dataset for a given language pair.

Regarding percentages of titles and documents in the training data, more abstracts are available in comparison to titles. This aspect is due to the high number of documents in Scielo that have their abstract translated to other language but not their titles, such as document S0874-48902010000300006 cited above. This is certainly a good feature of our corpus, given that previous parallel corpora of biomedical publication were restricted to MEDLINE titles [7, 3]. On the other hand, the monolingual datasets are mainly composed of titles, due to the same reason stated above, i.e., the existence of many articles whose titles were been translated to other languages. Finally, we officially released only parallel datasets that include English in the language pair. However, there are some documents which are available for other language pairs, such as ES/PT, ES/FR and FR/PT, as illustrated in Figure 2.

4 Future work

As further work, we could evaluate the contribution of additional, in and out of domain, available corpora to improve MT results. We plan to make the training set available in community challenges (e.g. ACL WMT'16) so the research community can experiment with additional translation methods. Finally, it would be interesting to use this corpus in a task that could be used in a practical context and I have plans to explore the corpus for MT experiments, including implementation of MT algorithms in HANA.

References

[1] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

[2] D. C. Comeau, R. Islamaj Doan, P. Ciccarese, K. B. Cohen, M. Krallinger, F. Leitner, Z. Lu, Y. Peng, F. Rinaldi, M. Torii, A. Valencia, K. Verspoor, T. C. Wiegers, C. H. Wu, and W. J. Wilbur. Bioc: a minimalist approach to interoperability for biomedical text processing. *Database*, 2013, 2013.

[3] A. Jimeno Yepes, E. Prieur-Gaston, and A. Neveol. Combining medline and publisher data to create parallel corpora for the automatic translation of biomedical text. *BMC Bioinformatics*, 14(1):146, 2013.

[4] K. Kirchhoff, A. M. Turner, A. Axelrod, and F. Saavedra. Application of statistical machine translation to public health information: a feasibility study. *Journal of the American Medical Informatics Association*, 18(4):473–478, 2011.

[5] P. Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[6] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open source toolkit

for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[7] J. A. Kors, S. Clematide, S. A. Akhondi, E. M. van Mulligen, and D. Rebholz-Schuhmann. A multilingual gold-standard corpus for biomedical concept recognition: the mantra gsc. *Journal of the American Medical Informatics Association*, 22(5):948–956, 2015.

[8] M. Neves, A. J. Yepes, , and A. Névéol. The scielo corpus: a parallel corpus of scientific publications for biomedicine. In *Language Resources and Evaluation Conference, 23-28 May 2016, Portoro, Slovenia (under review)*, 2016.

[9] P. Pecina, O. Duek, L. Goeuriot, J. Haji, J. Hlavov, G. Jones, L. Kelly, J. Leveling, D. Mareek, M. Novk, M. Popel, R. Rosa, A. Tamchyna, and Z. Ureov. Adaptation of machine translation for multilingual information retrieval in the medical domain. *Artif Intell Med*, 61(3):165–85, Jul 2014.

[10] C. Wu, F. Xia, L. Deleger, and I. Solti. Statistical machine translation for biomedical text: Are we there yet? *AMIA Annual Symposium Proceedings*, pages 1290–1299, 2011.

Analyze Genomes: A Cloud Platform Enabling On-Site Analysis of Sensitive Medical Data

Matthieu-P. Schapranow, Cindy Perscheid
Hasso Plattner Institute
Enterprise Platform and Integration Concepts
August–Bebel–Str. 88
14482 Potsdam, Germany
{schapranow|cindy.perscheid}@hpi.de

Abstract

Nowadays, ever growing amounts of medical data can be produced in a short period of time and need to be analyzed to acquire insights. Cloud computing has gained in importance, as its shared computing resources are accessible by everyone. However, the analysis of data in the cloud requires its transfer from local systems to shared cloud resources, which involves a significant amount of time depending on the data size. In addition, legal requirements due to data privacy can pose obstacles for using cloud systems especially in the area of life sciences, as projects here often deal with sensitive patient data.
With our Analyze Genomes cloud computing platform, we aim at addressing these requirements by integrating existing decentralized computing resources to form a federated in-memory database system. It enables research facilities to consume managed software services whilst sensitive data remains stored and processed on their local hardware resources.

1 Project Idea

With the ongoing technical advances in laboratory equipment, more and more fine-grained biological and diagnostic data is generated in a shorter period of time. Due to their increasing volume, these data sets cannot be analyzed manually any longer but instead require computational analysis workflows. In the scope of our Analyze Genomes project, we have set up a federated cloud platform that provides such analysis workflows and additionally combines the results with scientific data from distributed data sources [1, 3].
Nowadays, the use of cloud services requires users to transfer local data in advance to the shared computing resources of the cloud service provider. On the one hand, with increasing amounts of data being produced, loading up data to shared resources results in a significant delay in processing and analysis. On the other hand, most small- and mid-sized labs have to outsource computational analysis of their experiment results to optimize workflows. Furthermore, international collaborations between research centers are important for finding new scientific insights. However, these take place only to a limited extend and face various IT challenges nowadays, e.g. heterogeneous data formats and requirements imposed by legal and privacy regulations.

2 Current Project Status

In the past Future SOC lab periods, we focused on an approach we call Federated In-Memory Database (FIMDB) to leverage the usage of our cloud services at local clusters of research facilities [2]. We have defined a cloud setup integrating decentralized computing resources from a research facility whilst we as a service provider manage algorithms and applications that are executed on our remote system. This way, sensitive data remains at local sites. In the following, we document our current progress in and experience gained from developing a FIMDB system.

2.1 Reorganization of the Worker Framework File Structure

The changes made to our infrastructure so far to include the cluster of a Berlin research facility involved changes to the database landscape and user administration. However, with data and computing resources being potentially located outside our own computing facilities, we need to extend the internal worker framework structure and mechanisms as well.

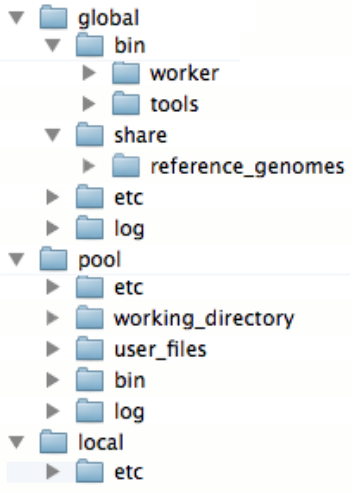


Figure 1: Adapted file structure for the Analyze Genomes cloud platform. Files are now located according to be shared globally across all computing nodes, within only a pool of them, or with no one else.

The worker framework consists of: binaries and executables for the worker routines, third-party tools, temporary directories with intermediate processing results, and additional files required for execution. All this was located in one file system shared across all nodes of our computing cluster using Network File System (NFS).
By including computational resources from external sites, however, local data for analysis must reside at the research facilities' computing clusters all the time. In addition, it must remain shared within the computing cluster of the research facility to enable distributed processing. This extended setup demanded for a change in our current file system structure as outlined in the following.
Instead of one single location for all files, we reorganized our file system structure into the three categories of directories *global*, *pool*, and *local* as depicted in Figure 1.

Global directories are shared across all computing nodes, including clusters from individual research facilities and service providers. It contains amongst others the subdirectories `bin`, `log`, and `share`. `bin` contains all executable files of our worker framework, e.g. it contains executables of the workers themselves and third-party binaries that are used for analysis pipelines. In addition, most of the third-party tools require other data files, e.g. for reference genomes, for their execution, which are located in the `share` subdirectory. Any logs written by workers during execution of a pipeline across all computing nodes will be written into log files located in the `log` subdirectory.

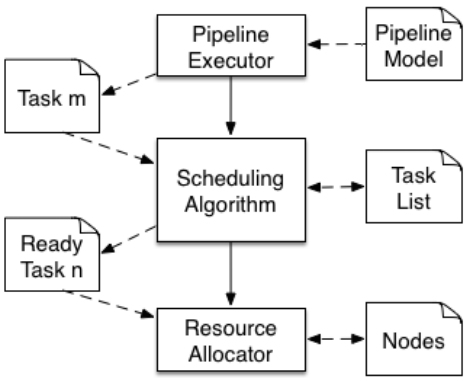


Figure 2: Adapted scheduling architecture of the Analyze Genomes cloud platform. Coordination of pipeline execution now splits up into the three components of Pipeline Executor, Scheduling Algorithm, and Resource Allocator.

Pool directories are shared across compute nodes that belong to a certain pool, e.g. a research facility's computing cluster. At the moment, each research facility can only belong to one pool at the same time. The directory also contains a `log`, `bin`, `working_directory`, and `user_files` directory. The `log` directory contains all logs written by workers during execution of a pipeline that is executed exclusively with computing resources of that pool. The `bin` directory contains all files that are required for execution of the analysis pipeline but must reside at the research facility's computing resources, e.g. if a research facility wants to use in-house tools. The `working_directory` contains all files that are created during pipeline execution, with subdirectories for the corresponding pipeline runs. The `user_files` directory contains all files that are uploaded by users, e.g. experiment results that must be analyzed with our pipelines.

Local directories are only accessible by a single compute node and not shared with any other node. For example, it contains configuration files for node-specific paths and parameters.

2.2 Extension of Scheduling Procedures

The scheduling routines applied within our Analyze Genomes worker framework assign tasks to any worker that is not working on any task. However, with the extended system setup and its requirements imposed by incorporating computation nodes of external research facilities, we need to refine our scheduling mechanisms accordingly.
Figure 2 depicts our extended scheduling archi-

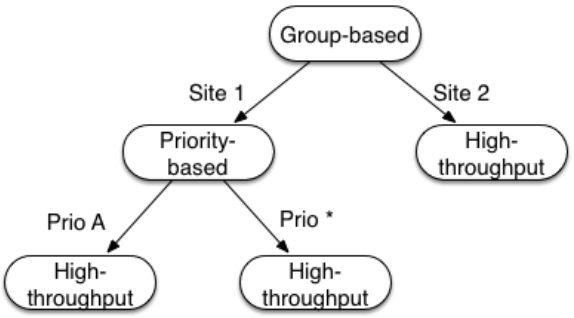


Figure 3: Scheduling modules can be combined hierarchically as desired, e.g. on a first level based on membership in a particular group, on a second level according to priority, and on a third level according to task throughput.

tecture. Instead of one single scheduler component taking care of the complete pipeline execution, we have split it up into the three components of Pipeline Executor, Scheduling Algorithm, and Resource Allocator.

The Pipeline Executor takes care of the overall pipeline execution. This component reads the model description of a pipeline stored as XML Process Definition Language (XPDL) format and transforms it into an executable graph structure of task objects [4]. Once a task is ready for execution, e.g. because all its preceding tasks have been finished, the Pipeline Executor creates the corresponding task object. Such a task object contains parameter information, e.g. if and what reference genome must be provided, and a wait count indicating how many tasks must be finished before this one can be executed.

Task objects that are ready for execution are forwarded to the Scheduling Algorithm. This component comprises the overall scheduling logic. Before the extension of our computing landscape by clusters of external research facilities, there existed the most common scheduling routines, e.g. shortest- or longest-task-first and first-in-first-out [5]. The adapted landscape architecture requires the extension of our scheduling policies, as we need to ensure that exclusively specific pools of computing nodes execute specific pipelines. Therefore, we split up the available scheduling routines into modules called *group-based*, *priority-based*, and *high-throughput*.

These can be combined hierarchically as shown in Figure 3. For example, on the first level tasks are scheduled according to their membership in a particular group to let a user execute his tasks at the computing nodes of his research facility only. On the second level, all tasks of the same group can then be scheduled according to their priority, e.g. to prefer an important user whose tasks should be favored for execution above other members of that group. On the third level, all tasks with the same priority can be scheduled according to their throughput to deal with high system load, e.g. shorter tasks are favored over tasks that require more execution time to deal with high load. Tasks that are scheduled next for execution are provided to the Resource Allocator component. This component maintains a list of all running and idle nodes, including their associations to specific groups and users. Once a node is idle, it sends a message via User Datagram Protocol (UDP) to the Resource Allocator, which identifies a task that can be executed by that node from its task list. If properties of task and node match, i.e. the node is associated to the same group as the task, the Resource Allocator assigns the task to this node. If there is no matching task, the node remains idle until a new matching task is available and assigned by the Resource Allocator.

3. Conclusion

Sharing knowledge is the key for reproducible research cooperations. While nowadays this is hampered not only by legal restrictions but also technical challenges, we aim at designing a system that fulfills technical requirements to enable research sites using cloud services whilst their sensitive data remains physically on local resources. By using our adapted Analyze Genomes FIMDB platform, research sites can use their own technical infrastructure for analyzing their data without having to set up tools and pipelines for analysis. By that, delays in processing the data sets due to either setting up new analysis procedures or transferring the data to another site are avoided.

In this report, we outlined selected extensions to our Analyze Genome's worker framework in order to match requirements that accompany the incorporation of computing nodes from external research facilities into Analyze Genomes' landscape. This new setup in data sharing required us to completely rethink the file structure underlying the worker framework and led to a clear distinction between files that are shared globally, across a pool of computing nodes, and locally only. We have also outlined adaptations to the scheduling component and policies of the worker framework, to guarantee that all data owned by a research facility is stored and processed at their own site.

References

[1] H. Plattner and M.-P. Schapranow, editors. *High-Performance In-Memory Genome Data Analysis: How In-Memory Database Technology Accelerates Personalized Medicine.* Springer-Verlag, 2014.

[2] M.-P. Schapranow et al. A Federated In-Memory Database System For Life Sciences. *Business Intelligence for the Real Time Enterprise (BIRTE) Workshop at VLDB*, 2015.

[3] M.-P. Schapranow, F. Häger, and H. Plattner. High-Performance In-Memory Genome Project: A Platform for Integrated Real-Time Genome Data Analysis. In *Proceedings of the 2nd Int'l Conf on Global Health Chall*, pages 5–10. IARIA, Nov 2013.

[4] R. M. Shapiro. XPDL 2.0: Integrating Process Interchange and BPMN. *Workflow Handbook*, pages 183–194, 2006.

[5] A. Tanenbaum. *Modern Operating Systems*. Pearson Education, Inc., 2009.

Towards Predictive Analytics for Dynamic Evolutionary Optimization

Wei Cheng
SAP Innovation Center Network, Potsdam, Germany
wei.cheng@sap.com

Julia Jordan, Bernd Scheuermann, Jörn Weber
Hochschule Karlsruhe, University of Applied Sciences, Karlsruhe, Germany
bernd.scheuermann@hs-karlsruhe.de

Abstract

This paper proposes and outlines approaches to using SAP HANA to expedite the search process of Evolutionary Algorithms addressing combinatorial optimization problems in dynamically changing environments. Novel prediction methods are presented targeting PAL with the aim of anticipating forthcoming dynamic change events based on the evaluation of historical records of previous change events and of optimization knowledge persisted in HANA. The goal is to be better prepared for dynamic changes and to react quicker. The paper also describes strategies for the extraction and pre-processing of optimization knowledge to be stored in HANA. Furthermore it is outlined how knowledge can be suitably selected from HANA and then injected into the running optimization.

1 Introduction

A wide range of industrial applications yield optimization problems which are known to be NP-hard problem (see, e.g., [5] for a concise introduction to the intractability of optimization problems). Such problems include, e.g., the Vehicle Routing Problem (VRP) [15], the Traveling Salesperson Problem (TSP), the Knapsack Problem [7] or many other problems in production, warehouse and transportation logistics. Considering such NP-hard combinatorial optimization problems, exact algorithms guarantee to find the optimum, but it is widely agreed that in the worst case, the search requires exponential time. In this case, the only affordable option is to apply approximate algorithms, also called heuristics, which search for good solutions, however, they cannot guarantee to find the optimum. The focus of this paper is set on bio-inspired heuristics called Evolutionary Algorithms [6] which mimic the principles of evolution and natural selection.

In typical real-world scenarios, the optimization problem must not be considered as static. Instead different aspects like the objective function, the size of the problem instance or constraints may be subject to changes over time. If any of such uncertainty is taken into account during the optimization process then the optimization problem is called *dynamic*. A straightforward approach to tackling such problems would be to consider each change as the arrival of a new optimization problem and to restart optimization from scratch. This may be a valid approach in some scenarios, e.g. when changes are severe or during initial iterations of the optimization process. In the majority of cases, however, the time for re-optimization is rather limited and one is interested in re-using existing optimization knowledge to quickly react to and to recover from a dynamic change.

This paper proposes to exploit the strengths of SAP HANA to expedite the search process in evolutionary algorithms, in particular addressing optimization problems in dynamically changing environments. Although being very successful in a wide range of industrial-strength optimization scenarios, typically evolutionary algorithms suffer a major drawback: they "forget" their search history. Therefore, this paper proposes an approach to resolve this weakness by designing and developing novel techniques for bio-inspired metaheuristics guided through extensive amounts of optimization knowledge managed by SAP HANA. Such optimization knowledge includes, e.g., historical logs of visited search areas, environmental data, and recorded change events. The aim is to enable the optimizer to predict forthcoming change events by using HANA and its Predictive Analysis Library (PAL). Furthermore, the goal is to make the optimizer better prepared for the prospected changes, to quicker respond to such

changes and to easier recover from their impact.
The remainder of the paper is structured as follows: Section 2 briefly introduces to dynamic evolutionary computation and the dynamic Knapsack Problem, which is further employed for illustration purposes. Section 3 outlines the design of the framework architecture and the implementation of the proposed evolutionary optimizer based on HANA. Subsequently, a range of prediction strategies as well as knowledge extraction and injection approaches are discussed in sections 4 and 5. Concluding remarks and an outlook on future work are provided in Section 6.

2 Dynamic Evolutionary Optimization

This section briefly introduces evolutionary optimization using the Knapsack Problem as an example of a complex combinatorial problem covering its static and dynamic variant.

2.1 Evolutionary Optimization

Prior to introducing the Evolutionary Algorithm, the static Knapsack Problem shall be defined and henceforth be used to exemplify the strategies proposed in this paper. In its static variant, the 0/1 Knapsack Problem is described by a set of n items of weight w_i and value v_i where $i \in \{1, \ldots, n\}$. A candidate solution of the problem is a subset of items $X = \{x_1, \ldots, x_n\}$ with $x_i \in \{0, 1\}$ indicating if item i is included in the knapsack which has a capacity of C. The goal is to maximize the total value of items included in the knapsack such that the sum of their weights is less or equal to the knapsack capacity: Maximize $f(X) = \sum_{i=1}^{n} v_i x_i$, subject to $\sum_{i=1}^{n} w_i x_i \leq C, x_i \in \{0, 1\}$.
As the Knapsack Problem is known to be $NP-hard$, evolutionary algorithms are one common heuristics to search for near optimal solutions. Inspired by the principles of natural evolution, the main idea behind evolutionary optimization is to represent solutions of an optimization problem as a set of individuals called population. The size of the population shall be denoted as m. An individual $j \in \{1, \ldots, m\}$ is encoded in a chromosome X_j representing the individual's genotype. In the case of the Knapsack Problem, individual j is encoded as n-bit chromosome $X_j = (x_{1j}, \ldots x_{nj}\}$ with $x_{ij} \in \{0, 1\}$, where $x_{ij} = 1$ means that in individual j item number i is contained in the knapsack, and $x_{ij} = 0$ otherwise. The Evolutionary Algorithm aims to incrementally improve on the set of individuals by mimicking the principles of natural selection, recombination, mutation and survival of the fittest (cf. [6] for a detailed introduction). One possible implementation is given below:

s1 Initialize population of m chromosomes.

s2 Evaluate the population: Calculate the fitness $f(X_j)$ of each individual j.

s3 Select 2 chromosomes as parents from the population.

s4 Form two children (offspring) with crossover probability p_c by recombining their chromosomes at a random point (otherwise the offspring are copies of their parents).

s5 Mutate the offspring by flipping every bit with mutation probability p_m.

s6 Evaluate the newly created offspring.

s7 Replace the current population with offspring.

s8 If no stopping condition is met goto step s3.

For the Knapsack Problem, some chromosomes may potentially violate the knapsack capacity constraint. In such cases, one should implement suitable repair algorithms or introduce penalty costs in order to enforce (or to foster) the search for feasible solution (not further outlined here, for the sake of brevity).

2.2 Dynamic Optimization Problems

In many optimization problems, different aspects like objective function, the size of problem instance or constraints may be subject to changes over time. If any of such uncertainty is taken into account during the optimization process then the optimization is called dynamic. Accordingly, the dynamic knapsack problem extends its static counterpart by introducing time-dependent variance: capacity $C(t)$, weights $w_i(t)$ and values $v_i(t)$ are considered as dynamic over time t. The goal is to maximize $f(X, t) = \sum_{i=1}^{n} v_i(t) x_i$ at any time t, subject to $\sum_{i=1}^{n} w_i(t) x_i \leq C(t), x_i \in \{0, 1\}$.

2.3 From Static to Dynamic Optimization

For dynamic optimization it is interested in reusing existing optimization knowledge to quickly react to and to recover from a dynamic change. The idea is to re-use information from the previous environment to accelerate optimization after a change. This implies that algorithms in dynamic environments are no longer focused on locating a stationary optimal solution. Rather they need to remain flexible to be able to track the movement of peaks through space and time. Or they create, maintain or control a certain degree of diversity such that solutions are spread across the search space which increases the likelihood of having solutions in the vicinity of the new optimum.

Starting with early work by [4], evolutionary algorithms have ever since been the most common approach to solving dynamic optimization problems [3]. Rohlfshagen [10] defined a problem based on a class of 0/1 dynamic knapsack problem, which are generated by a small set of real-valued parameters. Branke et al. [2] analyzed different representations on dynamic multi-dimensional knapsack problem. Simões [13] applied diversity-maintaining techniques and memory strategies to evolutionary algorithms for the dynamic knapsack problem.

3 Design Overview

3.1 Framework Architecture based on SAP HANA

The proposed approach is embedded into a project working towards a programming and execution framework for bio-inspired optimization exploiting the in-memory database of SAP HANA as knowledge store and its analytical engines to explore and exploit this knowledge. An overview is provided in Figure 1.

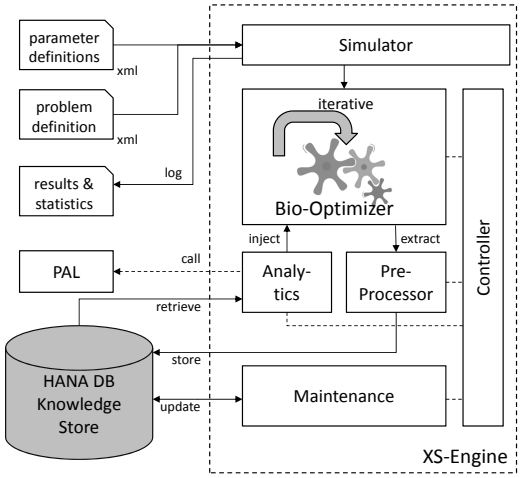


Figure 1: Architectural Overview

The framework is being implemented as native application on the SAP HANA XS Engine. A *simulator* reads the definition of the problem instance and parameter definitions of the optimization algorithm, produces a configurable online stream of change events and forwards the information to the *bio-optimizer* (here implemented as Evolutionary Algorithms). The *pre-processor* is responsible for extracting this raw data from the optimizer, pre-computing it and routing the results to the associative memory residing in SAP HANA. The *analytics* module evaluates the knowledge store and aims at maintaining solution diversity and identifying promising albeit unexplored areas in the search space. Here, the capabilities of PAL [11] may be exploited to analyze historical entries in the knowledge store thereby predicting future movements of optima. Housekeeping tasks are performed by the *maintenance* module which could execute age-dependent or quality-dependent deletions of solutions and to automatically re-evaluate solutions at dynamic changes. Finally, the *controller* instance monitors and coordinates the interplay of the modules described above. For a more detailed description of this framework refer to [12].

3.2 Implementation Outline for Dynamic Optimization

Figure 2 illustrates the sequence of a standard Evolutionary Algorithm (EA) as described in Section 2 (s1-s7). If the stopping condition is not met, the level of diversity is checked and if necessary random individuals are inserted (s8-s10) in order to maintain diversity.
The EA is extended by associative memory (black, implementing the *pre-processor* and knowledge store in Figure 1) and predictive analysis (gray, implementing the *analytics* part of the framework in Figure 1). At the beginning, the prediction method and the knowledge memory are initialized (m1, p1). Afterwards all generation extraction strategies are executed (m2, see Section 5). If the level of diversity decreases below a threshold, or if a change is detected that was not anticipated (thus the predictive part did not prepare the algorithm for this change), the old and new environment are compared and suitable individuals from the memory are reinserted into the EA (m3-m5).
The predictive part of the algorithm is organized in a loop running in parallel to the EA. First the existing data is evaluated (p3) and the prediction model is updated (p5). Next, the predictive analyses are performed (p6, see Section 4). Afterwards the algorithm returns to step p3. In case an upcoming change is predicted, steps m3-m5 and s8-s10 are triggered to prepare the EA based on the anticipation by injecting suitable solutions (or sub-populations) to predicted search areas. Whenever the algorithm detects an actual change, information on the EA's performance is fed back to the predictive part (p4) in order to measure predictive accuracy.

3.3 Data Modeling using SAP HANA

Figure 3 illustrates an exemplary excerpt of an associative memory. In Table *CapacityByGeneration* information on the capacity of the knap-

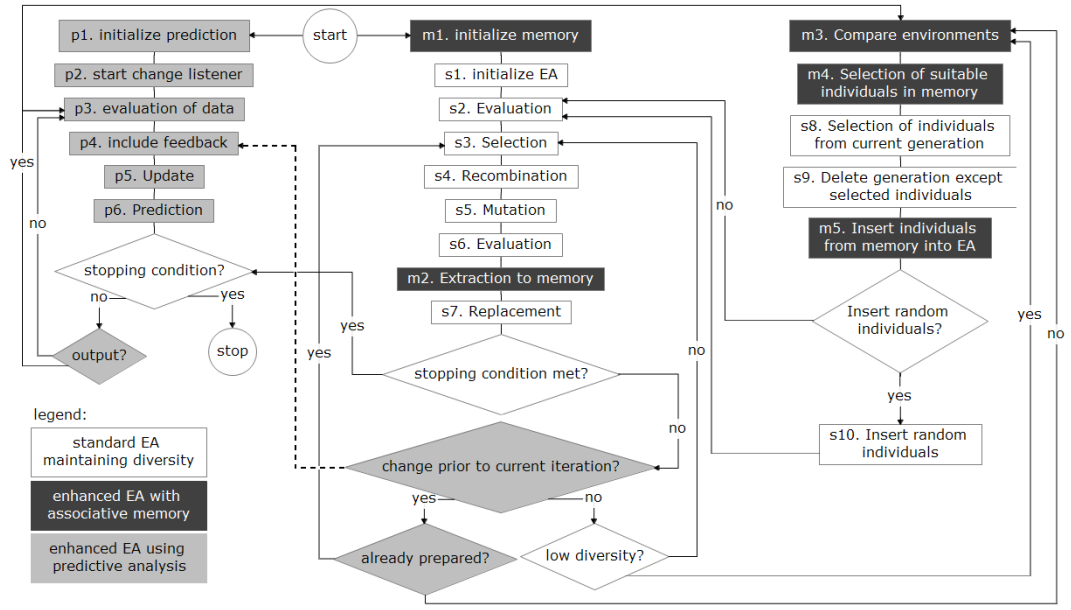


Figure 2: flow diagram of a standard EA (white), extended by associative memory (black) and predictive analysis (gray)

CapacityByGeneration	
int	int
GenerID	Capacity
0	100
...	...

WeightValueByGeneration			
int	int	int	int
ItemID	GenerID	Weight	Value
0	0	20	18
1	0	8	56
...	...	...	...

IndividualsByGeneration			
int	int	String	int
GenerID	Individual	Genotype	Fitness
0	0	X10X0101	356
0	1	X10X0110	340
0	2	X10X0100	298
1	0	010X0101	370
...	...	...	...

Figure 3: Associative Memory Overview

sack is stored per generation. Correspondingly Table *WeightValueByGeneration* stores weight and value of each item per generation. The extracted individuals from step m2 (Figure 2) are stored in Table *IndividualsByGeneration*. For each individual its genotype and fitness value are stored. A reasonable encoding of such genotypical information in a HANA database table needs to be identified. Referencing each item by using foreign keys would result in a large number of rows and a lot of redundant information in column *GenerID* and *Individual*. Another way would be to use one column per item. However, this approach requires constant updates of the database schema every time the number of items increases. The table in Figure 3 uses one column with the genotype encoded in one string. The first character in the string references the first item and so on. 1 indicates that the respective item is included in the knapsack, and 0 otherwise. A flag X is used to express that the respective item is not contained in this problem instance (which is dynamic) at the given generation. This approach requires, that the amount of items – regardless of the question whether they are included in a particular generation or not – is defined in advance. Alternatively, if the genotype may be encoded as one column per item. Parent-offspring dependencies may be added to the associative memory as well (not visualized here).

4 Predictive Analytics in Dynamic Optimization

In dynamic environments, if an EA does not react to a change, the performance of the algorithm will decrease rapidly afterwards. However adapting to a new environment takes time. Predictive techniques can increase the performance of the EA. In that way the algorithm anticipates changes and prepares itself before the change, thus avoiding the decrease in performance when the change actually occurs [14]. The advantage of prediction lies in the ability of active preparation in contrast to passive reaction (e.g. when using solely memory schemes or maintaining diversity) [14].
This paragraph reviews work related to predic-

tive techniques in EA. A state of the art survey by Branke [9] presents an overview of benchmark problems, performance measures and optimization approaches for evolutionary dynamic optimization. The introduced optimization approaches cover maintaining diversity, memory approaches (implicit and explicit), multipopulation and prediction, including advantages and drawbacks of each approach compared to the others. Change detection is also addressed, which is a relevant basis for predictive techniques. For a recent review of literature especially focusing on prediction in EA's for dynamic environments refer to [14]. A mathematical overview of six types of changes are given in [8]. These types include: *small step*, *large step*, *random*, *chaotic*, *recurrent* and *recurrent with noise*. Working with HANA, PAL [11] is an essential part for predictive analysis. Its functionality is organized in six categories: *preprocessing*, *classification*, *statistics*, *regression*, *time series* and *clustering*.

Simões and E. Costa [14] state that "Simple linear regression analyzes the relationship between a response variable y and a single explanatory variable x", thus transforming a set of data into a mathematical relation. We plan to use regression to determine *when* the next change will happen, as demonstrated by Simões [14]. Let x be the number of a change $(1, 2, 3, ...)$ and the response variable y is the generation where that change occures. If for example every 20 generations the capacity C of the knapsack changes (increase or decrease), y will be $(20, 40, 60, ...)$ and the fourth change is predicted for generation 80.

Time series in PAL include e.g. exponential smoothing to calculate a moving average [11]. Exponential smoothing is suitable to anticipate the value of a variable. By using regression methods a fourth change of capacity C in generation 80 is predicted but it is unknown what the new value of C will be and whether C will increase or decrease. This can be done by calculating a moving average based on the past values of C. So by combining regression techniques with time series analysis we plan to anticipate changes in the parameters of the problem.

As introduced in Section 3.1 clustering algorithms help analyze the distribution of solutions across the solution space. For example individuals with the genotype $(1, 0, 0, 1, 1, 1)$, $(1, 1, 0, 1, 1, 1)$ and $(0, 0, 0, 1, 1, 1)$ only differ in their first two genes, meaning they are very close to each other, thus forming a cluster. One of the suitable clustering algorithms in PAL is DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [11]. DBSCAN starts with an arbirtrary, unvisited starting point and forms clusters based on a scan radius and a parameter dedicating the minimum number of points required to form a cluster [11]. So clustering facilitates the analysis of the distribution of individuals.

5 Extraction and Replacement Strategies

Extraction and replacement describe the processes of integrating the SAP HANA database memory to the EA, like visualised Figure 4.

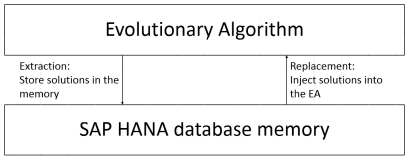


Figure 4: Integration of database to the EA

Extraction means taking individuals from the population of the EA and storing them in the memory. Thereby individuals may be replaced in the memory. The selection of these individuals is called update. Replacement means taking individuals from the memory and injecting them into the population of the EA. In order to maintain the population size some existing individuals have to be replaced. The memory can be integrated implicitly as a redundant representation in the EA, or it could be maintained explicitly as a separate memory component [9]. The motivation for integrating a memory to an EA is to compensate for the oblivion of previously found solutions by the EA. This may slow down convergence and favor diversity. By reusing information stored in the memory, it is also possible to reduce the computational time of the algorithm. Yang has shown that in [9].
Enabling the integration of an explicit memory also requires the definition of suitable strategies for updating the memory thereby addressing the following questions: Which individuals should be stored in the memory? How many solutions should be stored in the memory? When should the memory be updated? Which solutions in the memory should be replaced? Which solutions of the memory should be inserted into the EA in order to fasten computational time? Also the content of the memory has to be defined. Regarding the content, explicit memories can be divided in direct and associative memories. Direct memories only contain solutions from previous generations. In associative memories various types of information are stored together with solutions from previous generations, for example a list of the occurred environmental states, or the environment at which a stored solution exposed a good or very good fitness. Many other conceivable approaches can be found in [9].

In prior work, not only the fittest solutions have been used in the extraction. In order to maintain diversity, it can be profitable to also store inferior solutions [1]. For the replacement there are several approaches mentioned in prior work [1]: 1) Compute an *importance value* for each solution. The computation process can consider several values, for example the fitness value, the age and the diversity to the solutions in the population. Each component can be given a different weight. 2) Replace the solution which retains the maximum variance in the population after it is deleted. 3) Replace the solution which is the most similar one to the solution, which should be inserted, as long as the new one is fitter than the old. 4) Determine the two solutions with the minimum distance between each other in the population of the EA. Then replace the less fit solution. The above mentioned replacement strategies can also be used for updating the storage of the memory.
Subsequently, the selection of a solution (resp. an individual) is described in the context of different usages of the chosen solution, based on the database scheme of section 3.3. The necessary assumptions for an easily comprehensible example are: 1) The fitness values in the table *IndividualsByGeneration* are the values in the actual environment, calculated by the *analytics* module. 2) The actual generation number is five. 3) The only component used to compute the *importance value* is the fitness value of the solution. Regarding this assumptions the *importance value* of each solution equals its fitness. In a selection process the solution with the highest fitness value is selected for being stored in the memory. Referring to Figure 3 this is individual 0 of generation one. In an update and in a replacement process the solution with lowest fitness value is replaced. In Figure 3, this is individual 2 in generation 0.

6 Conclusion and Future Work

The work presented in this paper is part of a project which aims at using HANA for the purpose of enabling the optimizer to learn from the decisions of the past and to make better informed decisions in the forthcoming iterations of the optimization algorithm. Analyzing the optimization history shall enable the algorithms to discover as yet unexplored but promising regions in the search space. This paper set the focus to the currently running work which is devoted to developing approaches to using SAP HANA to expedite the search process of Evolutionary Algorithms addressing combinatorial optimization problems in dynamically changing environments. During the previous term a first version of a static Evolutionary Algorithm using HANA as knowledge store was implemented and tested in preliminary experiments. Current work is dealing with re-factoring and extending this algorithm for handling dynamically changing environments thereby adding prediction, extraction and injection methods as introduced in the previous sections.

References

[1] J. Branke. Memory enhanced evolutionary algorithms for changing optimization problems. *Congress on evolutionary computation CEC99*, pages 1875–1882, 1999.

[2] J. Branke, M. Orbayı, and Ş. Uyar. *Applications of Evolutionary Computing: EvoWorkshops 2006*, chapter The Role of Representations in Dynamic Knapsack Problems, pages 764–775. Springer, 2006.

[3] C. Cruz, J. R. Gonzalez, and D. A. Pelta. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Comput.*, 15:1427–1448, July 2011.

[4] L. Fogel, A. Owens, and M. Walsh. *Artificial intelligence through simulated evolution*. Wiley, 1966.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1983.

[6] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[7] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*. Springer, 2004.

[8] C. Li and S. Yang. A generalized approach to construct benchmark problems for dynamic optimization. In *Proc. of SEAL 2008*, volume 5361 of *LNCS*, pages 391–400. Springer, 2008.

[9] T. T. Nguyen, S. Yang, and J. Branke. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24, 2012.

[10] P. Rohlfshagen and X. Yao. The dynamic knapsack problem revisited: A new benchmark problem for dynamic combinatorial optimisation. In *Proc. of the Evo Workshops*. Springer, 2009.

[11] SAP. SAP HANA Predictive Analysis Library (PAL): SAP HANA Platform SPS 11, Document Version: 1.0 2015-11-25, 2015.

[12] B. Scheuermann and E. Weinknecht. On the potential of big data boosting bio-inspired optimization. Technical report, Future SOC Lab, HPI Potsdam, 2015.

[13] A. Simões and E. Costa. *Artificial Neural Nets and Genetic Algorithms*, chapter An Immune System-Based Genetic Algorithm to Deal with Dynamic Environments: Diversity and Memory, pages 168–174. Springer Vienna, Vienna, 2003.

[14] A. Simões and E. Costa. Prediction in evolutionary algorithms for dynamic environments. *Soft Computing*, 18(8):1471–1497, 2013.

[15] P. Toth and D. Vigo, editors. *Vehicle Routing: Problems, Methods, and Applications. 2nd Edition*. SIAM - Society for Industrial and Applied Mathematics, 2014.

ANALYTIC QUERIES ON TELENOR MOBILTY DATA

Ch. Niyizamwiyitira,
Blekinge Institute of Technology,
Karlskrona, Sweden,

L. Skold,
Telenor Sweden,
Stockholm, Sweden,

L. Lundberg,
Blekinge Institute of Technology,
Karlskrona, Sweden,

J. Sidorova,
Blekinge Institute of Technology,
Karlskrona, Sweden,
julia.a.sidorova@gmail.com

Abstract

This project focuses on the analysis of spatial data collected by Telenor Sweden. The project has two objectives: 1) developing the methods for spatial data analytics, and 2) finding an optimal technology to turn research prototype into scalable industrial application. The two main findings are as follows. Firstly, a smart marketing campaign based on the inferences from big spatial data has a high potential for the increase of Telenor's revenues. Based on the inference results from historical data, we recommend the categories of the users at whom the campaign should be targeted. Secondly, with a purpose to obtain recommendations for a feasible implementation, we have evaluated the performance of spatial queries on cluster and multiprocessor, comparing SQL and noSQL databases: PostgreSQL, MongoDB, and Cassandra.

1 Project Idea

Spatial Big Data has been accumulated by telecommunication operators and has a potential to be turned into business insights and then into revenues.

- The first objective was the development of new analytic methods or the application of the state-of-the-art solutions to get the answers to a number of questions, e.g. - *Which is the optimal user mix, given the infrastructure? - Which is the optimal location for a shop, if the mobility of different user categories is known?* At this stage a research prototype was developed.
- The next challenge was to find a scalable implementation to be able to efficiently analyze realistic-sized databases. We undertook a performance evaluation of different databases that handle big data on cluster and multiprocessor hardware.

Our case study has been completed on the spatial database covering user mobility in a medium-sized Swedish city during a week in 2015.

1.1 The *Tetris* Idea for Optimal Marketing

A major investment a telecom operator makes is the infrastructure and its maintenance, the revenues from which are proportional to the clientele's size. There are two major problems in the relation between clientele and infrastructure. Firstly, the infrastructure is of finite capacity, which makes it impossible to squeeze more than a certain number of subscribers into it and keep providing reliable service. Secondly, the subscribers use the network unevenly and some antennas are left under-used during certain time periods, which means a loss of potential revenues. The name *Tetris* comes from the famous computer game, where the objective is to fill the glass with figures of different shapes in an even manner, not leaving spaces.

The key observation is that clients have distinct mobility patterns: some are moving in areas with busy antennas, while others tend to stay in locations with under-loaded antennas. The marketing department of Telenor identified six different subscriber categories, which can be targeted individually by different marketing campaigns. This means that the subscriber mix can be modified in a planned and controlled way.

Tetris is an analytic strategy to find an optimal use of subscribers with the objective to increase the clientele's size without running into service failures due to overloading some antennas during some time periods. The *Tetris* strategy is based on linear optimization, a class of problems described with an objective function (in this case, maximize the number of sub-

scribers), and restrictions (in this case, finiteness of the infrastructure). Out of all possible solutions (all the possible mixes of clients), the best one is found, which yields the highest value for the objective function (the biggest number of users possible to be served with the present infrastructure). The most famous algorithm to solve this class of problems is the Simplex algorithm [1], which is also implemented in our solution [2].

1.2 Performance Evaluation of Trajectory Queries on Multiprocessor and Cluster

In this study, we evaluate the performance of trajectory queries on multiprocessor and cluster that are handled by

- PostgreSQL**,**
- MongoDB, and
- Cassandra.

Moreover, there are computationally distinct types of queries and they need to be assessed separately. The popular types of queries on spatial data are:

- *Distance query,* which returns the points that are located less than at some distance *l* from the user's location [3]*,* [4],
- *K-NN query,* which returns the closest points to the user's location [5], [6]*,*
- *Range query,* which returns the locations that are within the space range [3]*, and*
- *Region query,* which returns the region most likely to be passed by the user [3].

2 Used Future SOC Lab Resources

In single node installation, two types of servers are used,

1. *Hardware type 1* at Blekinge Institute of Technology with the following characterisations: Dell powerEdge R320, operating system: Ubuntu 14.04.3 LTS x86_64, RAM memory is 23 GB RAM, Hard disk size: 279.4GB 0 disk, the processor (Intel(R) Xeon(R) CPU E5-2420 v2) has 12 cores, each core is hyperthreaded into 2 cores, which results in 24 virtual cores.
2. *Hardware type2* at Future SOC Lab with the following characterisations: Fujitsu RX600S5, operating system: Ubuntu 13.04 LTS X86_64, the RAM memory is 1024 GB, Processor (4x Xeon X7550) has 32 cores, each core is hyperthreaded into 2 cores, this give 64 virtual cores.

In multiple node installation, a cluster of four nodes was used with all the nodes having the same features:

Hardware: Dell powerEdge R320, operating system: Ubuntu 14.04.3 LTS x86_64, RAM memory: 23 GB RAM, hard disk size: 279.4GB disk, processor (Intel(R) Xeon(R) CPU E5-2420 v2) has 12 cores, each core is hyperthreaded into 2 cores, this give 24 virtual cores.

3 Findings

We have obtained the following results:

1. We proposed a method called Tetris and demonstrated that a smart marketing campaign done with the help of Tetris analytics has a high potential for the increase of Telenor's revenues. Making inferences from data, we recommend at which user categories the marketing campaign should be targeted.
2. We have decided on the optimal implementation. The conclusions are based on the performance evaluation of spatial queries on cluster and multiprocessor, comparing SQL and noSQL databases: PostgreSQL, MongoDB, and Cassandra and considering four main types of spatial queries.

3.1 Marketing with Tetris

Under the assumption that the revenues are proportional to the clientele's size, the results demonstrate the possibility for a dramatic revenue increase, when *Tetris* is used compared to indiscriminative clientele expansion. In Figure 1, the revenue increase proportional to the antenna capacity is depicted. The blue line is the revenue increase, if the user mix in the clientele stays as it currently is. The red line is the revenue increase, if the marketing campaign is discriminative and relies on the Tetris' recommendations.

3.2 Not one size fits all!

The results of the performance evaluation of trajectory queries on multiprocessor and cluster (depicted in Figures 2-5) are summed up to the following:

1) Cassandra performs better than both MongoB and PostgreSQL to handle queries that do not have special geographical features, such as sphere, earth coordinates or other. An example of such an optimal query is the Region query, which involves time only.
2) Both Cassandra and MongoDB perform similarly on queries that have geographical features.
3) MongoDB has a built in function for spatial queries, and this speeds up the query response time.
4) Stratio's Cassandra Lucene Index plug into

Cassandra speeds up spatial queries.

In most cases we recommend the implementation with Cassandra.

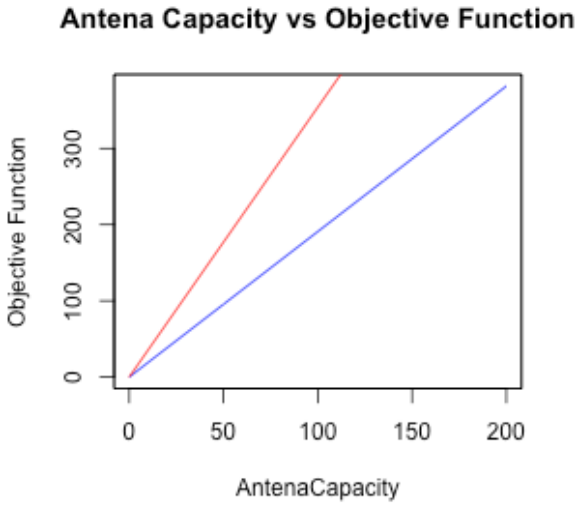


Figure 1: The revenue increases as the antennas' capacity is increased, based on marketing with Tetris (red) vs current mix of clients (blue).

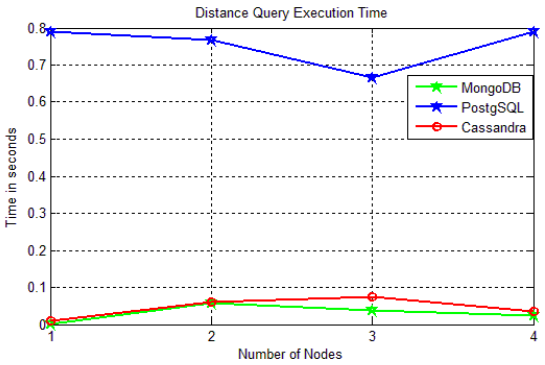


Figure 2: Distance query execution time.

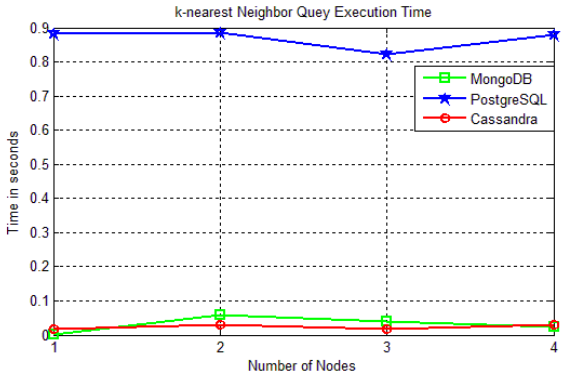


Figure 3: K-nearest neighbor query execution time.

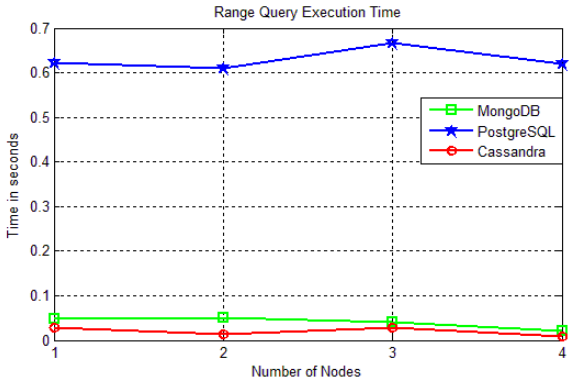


Figure 4: Range query execution time.

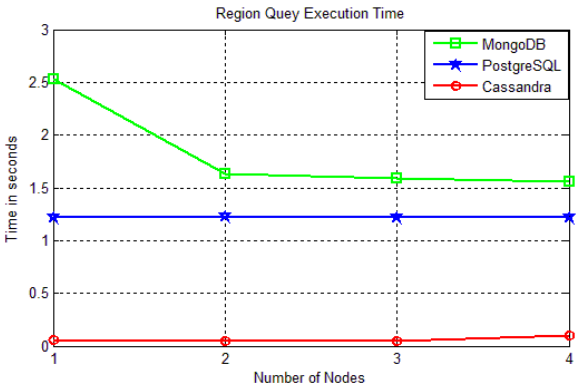


Figure 5: Region query execution time.

4 Next Steps

With respect to the methods, a number of different other analytics is being developed in around ten ongoing individual research projects, to mention a few:

- refine the user categories based on mobility clustering,
- give recommendations for the hospital to where to drive ambulances in the morning and how to move them during the week, depending on where people are,
- Assess the client with respect to her impact on the network and calculate how financially justified the deployment of the network is in different geographical zones.

With respect to the test-bed for the query efficency, the following databases have not been included into the study and are of relevance:

- graph database (Neo4j),
- key-value based (Redis),
- SAP HANA, and
- Influx DB (for time-series data).

References

[1] G. B., Dantzig, & M. N., Thapa, Linear programming 1: introduction. Springer Science & Business Media. 2006.

[2] Optimization G. Gurobi optimizer reference manual. URL: http://www. gurobi. com. 2012.

[3] Y. Zheng, and Z. Xiaofang, eds. Computing with spatial trajectories. Springer Science & Business Media, 2011.

[4] N. Pelekis, & Y. Theodoridis. Mobility data management and exploration. New York: Springer, 2014.

[5] B., Rimantas, Ch.. Jensen, G. Karčiauskas, and S. Šaltenis. "Nearest neighbor and reverse nearest neighbor queries for moving objects." In Database Engineering and Applications Symposium, 2002. Proceedings. International, pp. 44-53. IEEE, 2002.

[6] E. Frentzos, K. Gratsias, N. Pelekis, & Y. Theodoridis. "Nearest neighbor search on moving object trajectories." In Advances in Spatial and Temporal Databases, pp. 328-345. Springer Berlin Heidelberg, 2005.

Research and Development of Ensemble Learning Techniques for SAP HANA

Sabrina Plöger
FH Dortmund
Emil-Figge-Str. 42
44227 Dortmund, Germany
sabrina.ploeger@fh-dortmund.de

David Müller
FH Dortmund
Emil-Figge-Str. 42
44227 Dortmund, Germany
david.mueller@fh-dortmund.de

Christoph M. Friedrich
FH Dortmund
Emil-Figge-Str. 42
44227 Dortmund, Germany
christoph.friedrich@fh-dortmund.de

Christoph Engels
FH Dortmund
Emil-Figge-Str. 42
44227 Dortmund, Germany
christoph.engels@fh-dortmund.de

Abstract

Ensemble methods (like random forests, quantile forests, gradient boosting machines and variants) have demonstrated their outstanding behavior in the domain of data mining techniques.

This project focuses on literature research and development of ensemble learning methods, in order to propose strong techniques to be considered for further extensions of the SAP HANA PAL library.

1 Project Idea

In the first five Future SOC Lab periods, the University of Applied Sciences and Arts Dortmund successfully addressed the topic *Data Mining on SAP HANA* with their projects *Raising the power of Ensemble Techniques* and *Performance Optimization of Data Mining Ensemble Algorithms on SAP HANA* [1][2]. The initial project idea was to compare different opportunities, which enable the usage of predictive analytical techniques on SAP HANA.

SAP is offering the Predictive Analysis Library (PAL), which contains more than 70 well-known algorithms in the fields of classification analysis, association analysis, data preparation, outlier detection, cluster analysis, time series analysis, link prediction and others [3].

Starting with basic comparisons between SAP HANA PAL algorithms and R algorithms, they proceeded with implementing own data mining algorithms in different languages on SAP HANA. The final programming result is a random forest implementation in C++, which was introduced in spring 2014. This kind of ensemble learning algorithm was not available in the PAL library in these days and thus it gave greater opportunities for analyzing data stored on SAP HANA. In comprehensive tests, the random forest implementation evinced itself as a strong and fast algorithm with convincing prediction results. [4]

The project idea of the recent and upcoming Future SOC Lab periods is to support the SAP PAL development team from Shanghai directly. In a first step, results of the preceding projects are utilized by delivering the random forest with additional documentations of the underlying concepts. In a second step, comprehensive research analysis of ensemble learning techniques is performed in order to support the PAL development team in the process of selecting and implementing ensemble methods for the PAL library.

Why Ensemble Methods?

Predictive statistical data mining has evolved further over the recent years and remains a steady field of active research. The latest research results provide new data mining methods, which lead to better results in model identification and behave more robustly especially in the domain of predictive analytics. Most analytic business applications lead to improved financial outcomes directly, for instance demand prediction, fraud detection and churn prediction [5][6][7][8][9][10]. Even small improvements in prediction quality lead to enhanced financial effects. Therefore, the application of new sophisticated predictive data mining techniques enables business pro-

cesses to leverage hidden potentials and should be considered seriously.

Especially for classification tasks ensemble methods (like random forests) show powerful behavior [11][12][13] which includes that

- they exhibit an excellent accuracy,
- they scale up and are parallel by design,
- they are able to handle
    - thousands of variables,
    - many valued categories,
    - extensive missing values,
    - badly unbalanced datasets,
- they give an internal unbiased estimate of test set error as primitives are added to ensemble,
- they are robust to overfitting,
- they provide a variable importance and
- they enable an easy approach for outlier detection.

What are Ensemble Methods?

The main idea of ensemble methods is to combine a set of models, in order to obtain a better composite global model, which can reach more accurate and reliable estimates or decisions than one single model. This base learning algorithm can be a decision tree or any other learning algorithm.

Generally, it can be distinguished between two kinds of ensemble methods. *Independent ensembles* on the one hand, consist of models which can be trained independently from each other. Thus, the construction can be easily parallelized in appropriate multi-core environments like SAP HANA. Examples for the independent ensemble methodology are bagging and random subspace ensembles. [14] On the other hand, the models of a *dependent ensemble* mutually influence each other and are therefore interdependent. Because each model uses the knowledge of the previous models to adjust its construction, the ensemble has to be trained sequentially. Famous representatives of dependent ensemble methods are for example adaBoost, stochastic gradient boosting and iterated bagging. [15]

2 Used Future SOC Lab Resources

For this project a SAP HANA instance with the latest PAL revision and access to the HANA AFL SDK is provided. Access to the development environment of the AFL library is necessary for the implementation and integration of ensemble techniques into the PAL library.

3 Findings

The main deliveries of this project are the handover of the random forest implementation and corresponding documentation as well as the results of the literature review on ensemble learning techniques. In this analysis, a selection of innovative and state-of-the-art ensemble algorithms are investigated and evaluated. Each ensemble method is considered with regard to its process, availability, application and performance.

4 Next Steps

The main objective of this project is to support the PAL team in perspective research and development tasks. The cooperation between the project team of the University of Applied Sciences and Arts Dortmund and the SAP HANA PAL team is terminated for one year and ends in September 2016. In the following, potential activities for the upcoming project period are listed.

Activity "Support in selecting ensemble methods for PAL integration"

In the recent project period comprehensive research was carried out to determine strong ensemble prediction models. The first activity of the upcoming period is to support the PAL team in the selection process of those models, which should be integrated in the PAL library. All analyzed methods will be presented and a recommendation on the model selection will complete this part.

Activity "Support in implementing the selected ensemble methods"

The second activity is to support the PAL team in implementing the selected models. It must be determined, for which models the implementation is accompanied and how this process can be supported effectively.

Activity "Identification of a more user friendly development environment"

In the earlier periods, the project team worked with a simple editor and without appropriate debugging opportunities, which makes coding inconvenient. For the next period it is important to set up a more user friendly development environment, if the project team is instructed to support the programming part as well.

Activity "Open research tasks"

There are still open research topics which were not considered in the preceding project periods. Thus, research on the following topics might be carried out:

Anomaly detection

Anomaly detection, also known as outlier detection, is a subarea of data mining. The goal is to identify

untypical or conspicuous data in a dataset. In practice, users are facing different problems by applying those methods, as for example choosing the right method and adjusting its parameters or dealing with sparse and bad labeled datasets. This activity comprises the identification of advantages and disadvantages of those methods, the determination of strong algorithms and their implementation on SAP HANA. [16]

Machine learning on sparse data

Many data mining methods do not work well on sparse data [17]. The goal is to carry out research on state-of-the-art solutions and to implement a selection of those algorithms on SAP HANA.

5 References

[1] C. Engels, C. Friedrich: "Proposal - Raising the power of Ensemble Techniques", Proposal to summer 2013 period at the HPI Future SOC Lab, 2013.

[2] D. Müller, C. Engels, C. Friedrich: "Proposal - Performance Optimization of Data Mining Ensemble Algorithms on SAP HANA", Proposal to summer 2014 period at the HPI Future SOC Lab, 2014.

[3] SAP AG: "SAP HANA Predictive Analysis Library (PAL) (document version: 1.0 – 2015-11-25)", 2015, URL: http://help.sap.com/hana/SAP_HANA_Predictive_Analysis_Library_PAL_en.pdf, accessed on 22.03.2016.

[4] D. Müller, S. Plöger, C. Engels, C. Friedrich, "Optimization of Data Mining Ensemble Algorithms on SAP HANA", Project report to summer 2015 period at the HPI Future SOC Lab, 2015.

[5] R. E. Banfield et al.: "A Comparison of Decision Tree Ensemble Creation Techniques", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 1, 2007.

[6] S. Benkner et al.: „@neurIST Infrastructure for Advanced Disease Management through Integration of Heterogeneous Data, Computing, and Complex Processing Services", DOI:10.1109/TITB.2010.2049268, IEEE Transactions on Information Technology in BioMedicine, 14(6), Pages 1365 - 1377, 2010.

[7] C. Engels: "Basiswissen Business Intelligence", W3L Verlag, Witten, 2009.

[8] C. Engels; W. Konen: "Adaptive Hierarchical Forecasting".Proceedings of the IEEE-IDACCS 2007 Conference, Dortmund, 2007.

[9] J. Friedman: "Computational Statistics & Data Analysis", Volume 38, Issue 4, 28 February 2002, Pages 367–378, 2002, URL: http://dx.doi.org/10.1016/S0167-9473(01)00065-2, accessed on 22.03.2016.

[10] G. Üstünkar et al.: "Selection of Representative SNP Sets for Genome-Wide Association Studies: A Metaheuristic Approach", DOI:10.1007/s11590-011-0419-7, Optimization Letters, Volume 6(6), Pages 1207-1218, 2012.

[11] L. Breiman: "RF / tools – A Class of Two-eyed Algorithms", SIAM Workshop, 2003, URL: http://www.stat.berkeley.edu/~breiman/siamtalk2003.pdf , accessed on 22.03.2016.

[12] L. Breiman: "Random Forests", 1999, URL: http://www.stat.berkeley.edu/~breiman/randomforests-rev.pdf, accessed on 22.03.2016.

[13] G. Seni, J. Elder: "Ensemble Methods in Data Mining", Morgan & Claypool, San Rafael, California, 2010.

[14] Z. Zhou.: "Ensemble Methods. Foundations and Algorithms" CRC Press, Hoboken, 2012.

[15] L. Rokach, O. Maimon.: "Data mining with decision trees. Theory and applications." World Scientific, Singapore, 2008.

[16] C. Aggarwal: "Outlier Analysis", Springer, New York, 2013.

[17] T. Hastie, R. Tibshirani, M. Wainwright: "Statistical Learning with Sparsity: The Lasso and Generalizations", Chapman & Hall/CRC Monographs on Statistics & Applied Probability, CRC Press, Florida, 2015.

Analyzing the Global-Scale Internet Graph at Different Topology Levels: Initial Graph Analysis

– Technical Report & Extended Abstract –

Benjamin Fabian
Institute of Information Systems
Humboldt University of Berlin
Spandauer Straße 1
10178 Berlin, Germany
bfabian@wiwi.hu-berlin.de

Georg Tilch
Institute of Information Systems
Humboldt University of Berlin
Spandauer Straße 1
10178 Berlin, Germany
tilchgeo@wiwi.hu-berlin.de

Abstract

We integrated traceroute data from global-scale mapping projects to generate comprehensive Internet maps at different abstraction. In the next phase of the project, we conducted initial graph analyses with respect to identifying important nodes before we aim to assess Internet robustness via simulations.

1 Internet Topology

The purpose of the Internet as a globe-spanning network is to enable connectivity among the billions of connected machines, i.e., each device should be able to communicate with every other device. Reliable information about the Internet topology is crucial to the development of effective routing algorithms, security purposes, robustness analyses, resilience management, and designing countermeasures against global surveillance.

This project [1] aims at developing methods for creating and analyzing a large integrated set of Internet graphs at the IP-interface level as the basis for subsequent examinations. Our analyses include the search for bottlenecks and weak points in the entire Internet topology as well as in the topological connectivity of individual firms and services.

2 Research Approach

This project aims at advancing the understanding of the Internet topology by integrating empirical data into a multi-leveled graph model. The main emphasis of our research project is placed on both generating and analyzing a combined global-scale Internet graph at different topological abstraction levels (i.e., IP-interface, Point-of-Presence PoP, Autonomous Systems AS).

From now on, focus is being placed on the analysis of the generated graphs.

3 Related Publications

The Institute of Information Systems at Humboldt University Berlin has been conducting research in the graph analysis domain for several years [1-9].

In particular, robustness analyses and vulnerability assessments of the Internet at AS-level have been conducted. Large-scale graph analysis has also been applied on the Bitcoin transaction network.

Further publications based on the current project are under development.

4 Project Plan

Our project requires powerful computation capabilities based on the large-scale memory and multi-core architecture of the HP Converged Cloud and the newly implemented SAP HANA Graph Engine. The project is structured in four phases.

The first phase of the project consisted of data acquisition and pre-processing. The second phase was concerned with extracting the graph at different granularities from the cleansed and combined raw data.

The third phase deals with the actual graph analysis of the extracted datasets, which is computationally expensive on such a massive scale. With the help of the computational power of HPI Future SOC Lab, we will probably be able to examine the centrality measures, clustering coefficients, shortest paths, and (strongly) connected components.

The (now updated) fourth phase consist of attack and failure simulations of parts of the graphs constructed in phases 1 and 2.

While the first, second and parts of the third phase are already completed, the remaining steps are also planned to be carried out on the resources provided by the HPI Future SOC Lab [10].

5 Project Status and Results

The project has successfully achieved major milestones of the first, second and parts of the third phase.

5.1 Phase 1: Data Integration

During the "IPv6 Launch" on June 6, 2012, major ISPs permanently enabled IPv6 for their services and since then, more and more traffic has been routed with the new system. This is the main reason why this work considers data collected during the timeframe of June, 7 – June 20, 2012 as observation period.

An overview of the traceroute data sources that have been integrated in our project is given in Table 1.

	iPlane	CAIDA	Carna	DIMES	RIPE Atlas	RIPE IPv6L
Size of raw data	45.9 GiB	86.2 GiB	17.8 GiB	30.7 GiB	20.3 GiB	30.5 GiB
Number of files	2,106	1,154	1	7	35	1
Number of records	264.6 mn.	203.3 mn.	67.0 mn.	21.0 mn.	20.9 mn.	10.3 mn.
Vantage points	299	56	266,604	783	4,780	56
Destination IPs	127,566	195.7 mn.	63.0 mn.	2.3 mn.	39	4,323
Number of traces	112.9 mn.	105.6 mn.	41.8 mn.	15.1 mn.	4.1 mn.	1.8 mn.

Table 1: Integrated Traceroute Data Sources

The acquisition and preprocessing of the data resulted in a final combined dataset with **281.5 million** unique traces for the observation period.

To our knowledge this is the largest and most diverse dataset in a traceroute-based topology discovery project so far and it establishes a thorough basis for the following graph extraction and analysis.

5.2 Phase 2: Graph Extraction

In Table 2, the fundamental statistics of the extracted graphs are shown (for the largest connected components, LCC).

	IP	Router	PoP	AS	ISP
Nodes	3,255,088	2,806,857	53,348	33,752	31,030
Edges	8,544,788	5,039,348	102,591	122,561	113,489
Avg. degree $\langle k \rangle$	5.2501	3.5907	3.8461	7.2624	7.3148

Table 2: Size Metrics of the Graphs (LCC)

5.3 Phase 3: Graph Analyses

We present some preliminary results of the graph analyses. The purpose of the average degree as a summary statistic of the degree distribution is to give an idea of how well the graph is connected in terms of neighbors. Degrees can also be seen as a centrality measure.

	IP	Router	PoP	AS	ISP
Minimum degree	1	1	1	1	1
Maximum degree	14,023	13,874	4,329	5,376	6,593
Avg. degree, ⟨k⟩	5.25011	3.590741	3.846105	7.262444	7.314792
Exponent, γ	3.22154	2.13352	2.41165	2.21073	2.23117
Standard Error	0.02885	0.00524	0.05981	0.01758	0.01698
k_{min}^{PL}	219	23	39	7	6

Table 3: Results for Degree Metrics

One of the most commonly studied properties of graphs is the degree distribution. Figure 1 plots the degree CCDF for all topology levels in one diagram. There is a strong visual evidence for the power-law relationships for the degree distributions on every level.

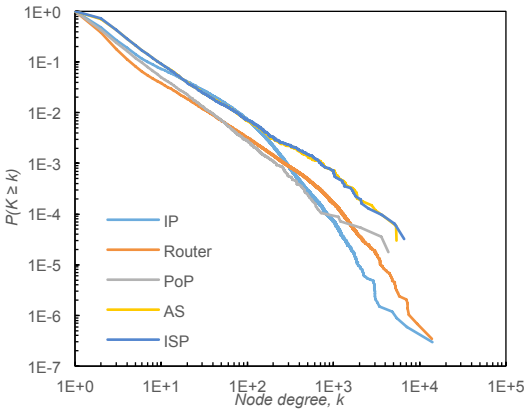


Figure 1: CCDF at different levels

Distance measures investigate graphs from a global "flow of information" perspective.

They are particularly interesting for traceroute-collected topologies since the measurement approach itself resembles path taking through the network.

The main drawback of distance metrics is that their calculation is very resource demanding. That is why some distance metrics could not be calculated so far for either the IP or the router graph, even on powerful hardware and more than a month of calculation time.

	IP	Router	PoP	AS	ISP
Avg. shortest path length, ⟨d⟩	-	-	4.2884	3.2060	3.1639
Diameter, d_{max}	46	45	15	7	7
Avg. Eccentricity, ⟨ε⟩	-	-	9.5710	5.5504	5.5476
Radius	-	-	8	4	4

Table 4: Results for Distance Measures

5.4 Use of Hardware Resources

The hardware provided by the HPI Future SOC Lab so far included three HP Converged Cloud Blades with 24 x 64-bit CPUs running at a frequency of 1.2 GHz on Ubuntu 14.04. Each of the three machines had 64 GiB memory and was equipped with 1 TiB HDD. This configuration permits an extensive parallelization of tasks.

The calculated results would not have been possible without the support of the HPI. For the intense calculations of the vulnerability analyses, the use of HPI Future SOC Lab resources is crucial.

6 Conclusion

The phases 1 and 2 of the project, data integration and graph extraction, have been completed. Also phase 3, initial graph analysis, has reached important milestones.

In future work, we aim to conduct the further steps of the project, in particular the vulnerability analyses of phase 4.

References

[1] B. Fabian and G. Tilch: Analyzing the Global-Scale Internet Graph at Different Topology Levels: Data Collection and Integration, *HPI Future SOC Lab Day Workshop & Report*, 2015.

[2] A. Baumann and B. Fabian, "How Robust is the Internet? – Insights from Graph Analysis," in *Proceedings of the 9th International Conference on Risks and Security of Internet and Systems (CRiSIS 2014)*, Trento, Italy, Springer, LNCS 8924, 2014.

[3] A. Baumann, B. Fabian, and M. Lischke, "Exploring the Bitcoin Network," in *10th International Conference on Web Information Systems and Technologies (WEBIST 2014)*, 2014, pp. 369–374.

[4] M. Lischke, B. Fabian: Analyzing the Bitcoin Network: The First Four Years, Future Internet 8(1), March 2016.

[5] B. Fabian, A. Baumann, and J. Lackner, "Topological Analysis of Cloud Service Connectivity," Computers & Industrial Engineering, vol. 88, pp. 151–165, October 2015.

[6] A. Baumann and B. Fabian, "Vulnerability Against Internet Disruptions – A Graph-based Perspective," Proceedings of the 10th International Conference on Critical Information Infrastructures Security (CRITIS 2015), Berlin, Germany, October 2015, Springer LNCS 9578.

[7] A. Baumann and B. Fabian, "Who Runs the Internet? Classifying Autonomous Systems into Industries," Proceedings of the 10th International Conference on Web Information Systems and Technologies (WEBIST), Barcelona, Spain, April 2014.

[8] A. Baumann and B. Fabian, "Towards Measuring the Geographic and Political Resilience of the Internet," International Journal of Networking and Virtual Organisations 12/2013; 13(4):365-384.

[9] M. Huth and B. Fabian: Inferring Business Relationships in the Internet Backbone, International Journal of Networking and Virtual Organisations, 2016.

[10] HPI Future SOC Lab. Available: https://hpi.de/forschung/future-soc-lab.html. Accessed 25 Mar 2016.

Automatic aggregation of training data for visual concept detection tasks

Christian Hentschel
Hasso-Plattner-Institut
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
christian.hentschel@hpi.de

Harald Sack
Hasso-Plattner-Institut
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
harald.sack@hpi.de

Abstract

Recent advances for visual concept detection based on deep convolutional neural networks have only been successful because of the availability of huge training datasets provided by benchmarking initiatives such as ImageNet. Assembly of reliably annotated training data still is a largely manual effort and can only be approached efficiently as crowd-working tasks. On the other hand, user generated photos and annotations are available at almost no costs in social photo communities such as Flickr. Leveraging the information available in these communities may help to extend existing datasets as well as to create new ones for completely different classification scenarios. In this project, we therefore aim to reliably identify photos relevant for a given visual concept category based on a large set of automatically crawled Flickr images.

1 Introduction

Visual concept detection refers to the ability of learning visual categories in order to automatically identify new, unseen instances of these categories. Typically, this task is approached as a supervised machine learning task: by using a reliably annotated dataset of example images separated into the categories the system should be able to recognize, a machine learns distinguishing features of the individual categories. Recently, approaches based on deep convolutional neural networks (CNN) have significantly improved over previous methods in terms of achieved classification precision. In two previous FutureSOC projects we have successfully evaluated the number of training images required for a CNN to outperform standard Bag-of-Visual-Words approaches [10] and compared the performance of a CNN trained on outside data with the performance of a CNN trained with no additional training data [3]. Our results show, that CNNs benefit from large amounts of training samples and even if pretrained on external data significant data is required in order to adjust a CNN to a new problem.

Acquiring manually annotated training data is a time-consuming (the ImageNet project[1] estimates the overall human annotation time required to support 40,000 categories by 10,000 individual images each to a total of 63 years), costly and error prone (due to the highly subjective nature of the task). On the other hand the World Wide Web provides huge data sources of annotated visual content almost for free. Photo sharing platforms such as Flickr host billions of user-generated images[2]. The community aspect has motivated millions of users to manually annotate their images with descriptive metadata such as image titles, tags and descriptions in order to increase the visibility of their photos or share their content with other users. Being able to exploit these information as training data not only would enable enlargement of existing datasets and categories by additional images. Considering the potentially unlimited vocabulary represented in user annotations a huge variety of additional categories could instantly be made available at almost no costs (even across a multitude of different languages).

However, a major drawback of these user annotations arising from the uncontrolled environment in which they are generated is that they need to be considered incomplete, highly subjective and not necessarily related to the visual content of the respective photo. This contrasts sharply with highly reliable image annotations required for learning visual concept classifiers and aimed at by initiatives such as ImageNet. Incompleteness – meaning that not all photos which depict a specific visual concept are actually annotated with a textual

[1] ImageNet: `http://image-net.org/`

[2] Flickr reports to host more than 6 billion photos: `http://blog.flickr.net/en/2011/08/04/6000000000/`

label identifying that concept – is usually of minor importance due to the potentially unlimited amount of annotated data available in photo communities. Subjective annotations and annotations with missing relevance to the depicted content, however, pose a major challenge when trying to retrieve images suitable to train a specific visual concept classifier.
In this project, we aim to automatically select photos – which are considered relevant to a given visual concept label by a majority of users – from a large collection of publicly available Flickr images. Our key assumption is that the majority of all users in a community shares a common (and thereby objective) interpretation of a visual concept which is reflected by a similar language used for annotation. We train a language model that captures the inter-author agreement and extend a query by using related terms (according to the language model). Training a language model is computationally costly. However, highly parallelized implementations exist, which can make use of multi-processor architectures as provided in FutureSOC lab.
Furthermore, we employ visual features extracted by convolutional neural networks in order to rerank images based on their visual similarity. We show, that a combined (textual as well as visual) similarity measure leads to better, i.e. more relevant images. CNN-based feature extraction makes use of an implementation adopted for GPUs such as the Tesla K20X as provided in FutureSOC lab.

2 Relevant Image Retrieval

2.1 Dataset

The Flickr platform provides a public API[3] to query their database and has already been used by many research activities in the past as it has become relatively easy to access a huge amount of photos and metadata. As an example, the MIRFLICKR-1M collection was published in [4] and consists of 1 million images crawled from Flickr. The selection of images has been made based on the Flickr *interestingness* score – a measure that aggregates factors such as clickthrough rate, user comments as well as users selecting an image as favorite. Additionally, the dataset provides authoritative metadata such as user tag data.
In [2] we have analyzed Flickr photos and user generated tags for relatedness. We've found out that annotations next to identifying the depicted content may also be used with an organizational or viewpoint defining purpose. Thus, even when an annotation explicitly mentions a visual concept this does not necessarily mean it is actually depicted. An algorithm that selects photos for usage as training data based on textual annotations should therefore be able to identify these photos as not being relevant for the respective visual concept. Here, we define a photo being relevant for a given visual concept if it depicts a clearly-visible version of the scene or object without any major occlusion.

[3]The Flickr API: `https://www.flickr.com/services/api/`

2.2 Community Language Model

For the aforementioned reasons, selecting photos solely based on the usage of the visual concept term within the annotations will likely fail to provide relevant photos. However, when considering the annotation context we assume that the majority of all users will use a similar vocabulary to describe the content. As an example, a photo depicting a sunset in many cases will also contain annotations such as "sea", "ocean", "clouds". When extending the query for "sunset" by these additional terms the retrieved photos will tend to exhibit higher relevance to the initial concept. Yet, manual creation of an extended query vocabulary per visual concept is prone to errors, subjective and most likely does not capture every relevant term.
In this paper, we therefore decided to learn annotation relationships based on contextual similarity immediately from the metadata corpus itself. This not only reduces a potentially error-prone manual query extension but also extracts additional terms based on the community users' applied vocabulary.
The authors in [8] present a neural network based approach to learn vector representations of single words, accordingly named *word2vec*. Training is performed in a completely unsupervised fashion – given a sufficiently large corpus, such as textual image annotations. A trained word2vec model allows to make highly accurate predictions about a word's meaning based on past contextual appearances. The output of a word2vec model is a vocabulary of all learned words and their respective vector representations. These can be used to compute the cosine similarity of words.
For our experiments, we have used the skip-gram implementation as provided in the gensim python package [9]. By training a word2vec model on the textual user annotations we enable extraction of similar terms given a visual concept label according to the language used by Flickr users. As an example, we have extracted the 10 most similar terms for the concept 'sunset':
Apparently, our initial assumption of the model being able to extract community specific related

sunset	dusk, sundown, sun, twilight, sunrise, cloud, silhouette, settingsun, nightfall, sky

terms holds: while terms such as 'sun', 'sunrise', 'cloud' and 'sky' could have also been manually selected as plausible query extension, the artificial term 'settingsun' can be only learned from the data itself.

2.3 Visual Re-ranking

As discussed in Sect. 1 we aim at increasing relevance by re-ranking candidate images based on their visual similarity. It has been shown that features extracted from the activation of a deep convolutional neural network which has been trained to separate individual visual concept categories on a large dataset can be reused and adapted to novel classification tasks [1, 11].
In order to obtain compact visual feature representations we make use of these findings by training a deep convolutional neural network on the ILSVRC-2012 dataset[4]. Model training is conducted using a Tesla K20 GPU provided at the Future SOC Lab in order to significantly reduce training time (total training time was about 5–6 days, depending on the number of iterations). Our implementation uses the Caffe CNN implementation [5] and extends from the successful architecture presented in [6].
In our experiments, we have used the vector of activities of the penultimate, fully-connected (seventh) layer (fc7) as feature descriptors, obtaining a 4,096 dimensional descriptor vector per image. We extracted the fc7-features for all images in the MIRFLICKR-1M collection. Using an NVIDIA Tesla K20 GPU, feature extraction took about 3 hours.
Wen compute the similarity of two candidate images by computing the cosine similarity of their respective layer-7 activity representations.

2.4 Experimental Setup

We test our approach on 10 selected visual concept categories. These categories comprise 8 object-level concepts ('airplane', 'bicycle', 'boat', 'bridge', 'car', 'dog', 'flower', and 'tiger') and 2 scene-level concepts ('beach' and 'mountain') and follow the categories chosen by the authors in [7]. We preprocess all tags by running lemmatization and stopword removal. Currently, we focus on English language only meaning that any other language is ignored for preprocessing. Analysis of the downloaded metadata corpus shows that users add an average of $|tags| = 12$ unique tags per photo ($\mu = 12.429, \sigma = 10.235$). We therefore set the window size for training our word2vec model to $w = |tags|/2 = 6$ words and ignore all words with a total frequency of $f < 5$. We train 300-dimensional feature vectors on the tag data and compute the $k = 20$ most similar terms for each visual concept label. Table 1 shows the concept labels and the (top-10) most similar terms according to our model.
The advantages of our model become easily visible: not only are relevant synonyms extracted (e.g. *airplane*: [aircraft, aeroplane, plane]) but also terms obviously *related* to the concept are selected (e.g. *beach*: [sand, ocean, shore]). Furthermore, frequent instances (e.g. *flower*: [dahlia, spiderwort, hyacinthaceae], *car*: [ford]) are extracted as well as translations of the original concept into different languages (e.g. *dog*: [chien]). Similarly, the model captures sub- and superclass relationships (e.g. *boat*: [fisherboat, sailboat, sailingships] and *tiger*: [flickrbigcats]) automatically from the dataset without having to extract them from an external knowledge base.
Based on these 20 most similar terms, we construct an extended query including the visual concept label. For each concept we then select those photos from the collection that best match the extended query assuming that images ranked higher are more likely to be relevant candidate images. We therefore rank images based on the number of query terms found in the respective tagset.
Visual re-ranking is applied to further increase the relevance of the top ranked images. We assume that the highest ranked image based on our extended query exhibits a high relevance for the visual concept and therefore re-rank the remaining images based on visual similarity to the top candidate. We compute the cosine similarity on the extracted deep feature representations as presented in Sect. 2.3.

3 Results

We compare our approach to a simple baseline algorithm (thus referred to as *baseline* hereafter), which selects photos based on whether or not the tagset contains the visual concept label (i.e. without any query extension). The number of candidate images based on this simple approach is considerably large. We therefore randomly sample $n = 200$ images for each visual concept category to evaluate the accuracy of the baseline method.
In order to evaluate a potential gain in accuracy by the individual steps, we have separately evaluated retrieval results based on the learned lan-

[4] ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012): `http://www.image-net.org/challenges/LSVRC/2012/`

Table 1: Visual concept labels and most similar terms according to skip-gram community language model

concept	similar terms
airplane	aircraft (0.90), aviation (0.88), aeroplane (0.86), plane (0.85), jet (0.85), airliner (0.84), jetliner (0.82), cockpit (0.81), regionaljet (0.78), planespotting (0.77)
beach	sand (0.78), ocean (0.70), shore (0.70), surf (0.69), wave (0.68), sea (0.67), zwemmen (0.64), kontikiinn (0.62), lowtide (0.62), capehenlopenstatepark (0.61)
bicycle	bike (0.88), cycle (0.88), cycling (0.85), citycycling (0.77), cyclist (0.77), bikelanes (0.77), bikelane (0.76), ridealong (0.76), citycycle (0.76), biking (0.76)
boat	sailing (0.79), ship (0.79), sail (0.77), moored (0.74), dock (0.74), yacht (0.73), fishingboats (0.73), sailboat (0.73), sailingship (0.72), port (0.72)
bridge	suspensionbridge (0.62), river (0.61), suspension (0.56), footbridge (0.56), swingbridge (0.53), building (0.52), brigde (0.52), riverhumber (0.51), barge (0.51), reka (0.51)
car	automobile (0.79), auto (0.76), sportscar (0.76), convertible (0.74), coupe (0.74), luxurycar (0.73), 6car (0.72), sedan (0.72), customcar (0.71), ford (0.71)
dog	puppy (0.89), canine (0.81), mutt (0.78), k9 (0.77), terrier (0.77), chien (0.76), interestingdogsposes (0.76), retriever (0.75), doggy (0.75), pup (0.74)
flower	bloom (0.74), daisy (0.72), flora (0.71), dahlia (0.70), spiderwort (0.69), hyacinthaceae (0.69), columbine (0.68), petal (0.68), flowercloseup (0.68), coneflower (0.67)
mountain	peak (0.73), hiking (0.69), mountainrange (0.68), snowcapped (0.68), valley (0.67), glacier (0.66), mountaineering (0.65), alpine (0.65), treck (0.64), gipfel (0.63)
tiger	flickrbigcats (0.61), pantheratigris (0.59), amurtiger (0.57), siberiantiger (0.56), cub (0.56), whitetiger (0.55), tigercub (0.55), sumatrantiger (0.55), bengaltiger (0.55), eagle (0.54)

guage model as well as based on additional visual re-ranking. Since both approaches generate ranked result set, we take the top $n = 200$ ranked candidates for evaluation. Evaluation results are reported as average precision scores corresponding to the area under the precision-recall-curve.
The results reported in Table 2 show the superiority of the proposed method. In general, the approach based on visual re-ranking of the results obtained from the trained language model outperforms the baseline approach as well as ranking based on textual features only. There are two major exceptions: While the results obtained for the category "airplane" based on the language model clearly outperform the baseline approach, we see a significant drop in the reported average precision when applying visual re-ranking. This is likewise true for "car" where the baseline approach even outperforms the textual model by 2%. When considering the top ranked image used as seed image for visual re-ranking for both classes we see that the image ranked highest according to our language model for the category "airplane" actually depicts an airport (although the number of found vocabulary tags indicate a high relevance for the "airplane" category). Visual re-ranking is therefore based on an airport image and fails to capture essential features of airplanes. Similarly, the highest ranked image for the category "car" actually depicts the rear light of an old car. Both misclassifications heavily decrease the achieved AP score and thus also affect the mean average precision score which is therefore slightly worse for the combination of language model and visual re-ranking. To avoid this in future, we plan to include more than just the top ranked photo for computation of visual similarities. A approach that we consider is to train a single-class SVM classifier based on the top-n highest ranked candidates according to our language model.

4 Future Work

The work presented here is only a first step towards exploitation of community photo data for visual concept classification. As discussed we aim to include further annotation data such as title, description and Flickr group information into our language model. Second, we aim to optimize parameters such as the number k most similar terms used to extend our initial query. Furthermore, we will train a classifier that considers the top-n candidate images to improve visual re-ranking. Finally, we will test the retrieved results in classification scenarios, i.e. we will evaluate the performance achieved by visual concept classifiers when trained on photos returned using our methods.

References

[1] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *International Con-*

Table 2: Comparison of proposed approaches for relevant image retrieval. Skip-gram is our approach based on the proposed community language model only. Skip-gram+vr denotes results obtained after additional visual re-ranking. Reported scores are average precision. Best results are marked in boldface.

concept	baseline	skip-gram	skip-gram+vr
airplane	0.457	**0.797**	0.237
beach	0.512	0.615	**0.812**
bicycle	0.476	0.850	**0.993**
boat	0.549	0.712	**0.936**
bridge	0.450	**0.611**	0.513
car	**0.621**	0.597	0.162
dog	0.758	0.885	**0.953**
flower	0.828	0.941	**0.980**
mountain	0.619	0.863	**0.977**
tiger	0.551	0.803	**0.959**
Mean AP	0.582	**0.765**	0.752

ference on Machine Learning, pages 647–655, 2014.

[2] C. Hentschel, H. Sack, and N. Steinmetz. Cross-Dataset Learning of Visual Concepts. In A. Nürnberger, S. Stober, B. Larsen, and M. Detyniecki, editors, *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, volume 8382, pages 87–101. Springer International Publishing, 2013.

[3] C. Hentschel, T. Wiradarma, and H. Sack. Comparison of feature extraction approaches for image classification. Technical Report Future SOC Lab report, 2015, Hasso-Plattner-Institute, 2015.

[4] M. J. Huiskes, B. Thomee, and M. S. Lew. New trends and ideas in visual concept detection. In *Proceedings of the international conference on Multimedia information retrieval - MIR '10*, page 527, New York, New York, USA, 2010. ACM Press.

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the ACM International Conference on Multimedia - MM '14*, pages 675–678, 2014.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9, 2012.

[7] X. Li, C. G. M. Snoek, and M. Worring. Learning Tag Relevance by Neighbor Voting for Social Image Retrieval. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, MIR '08, pages 180–187, New York, NY, USA, 2008. ACM.

[8] T. Mikolov, G. Corrado, K. Chen, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pages 1–12, 2013.

[9] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[10] T. Wiradarma, C. Hentschel, and H. Sack. Comparison of image classification models on varying dataset sizes. Technical Report Future SOC Lab report, 2015, Hasso-Plattner-Institute, 2015.

[11] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014, 13th European Conference*, volume 8689, pages 818–833. Springer International Publishing, 2014.

OntQA-Replica: Clustering with PAL and R for Ontology-Based Query Answering

Lena Wiese
Azadeh Amiri
Dorna Amiri
Research Group Knowledge Engineering
Institut für Informatik
Georg-August-Universität Göttingen
Goldschmidtstraße 7
37077 Göttingen
wiese@cs.uni-goettingen.de
azadeh.amiri@stud.uni-goettingen.de
dorna.amiri@stud.uni-goettingen.de

Abstract

The OntQA-Replica project aims to improve the performance of ontology-based query answering in distributed databases by employing a preprocessing procedure (including a clustering step and a fragmentation step): for efficient query answering, data records that are semantically related are grouped in the same data fragment. In this report we discuss applications of SAP HANA Predictive Analysis Library (PAL) and R for the clustering step.

1 Introduction

Flexible query answering [2] offers mechanisms to intelligently answer user queries going beyond conventional exact query answering. If a database system is not able to find an exactly matching answer, the query is said to be a failing query. Conventional database systems usually return an empty answer to a failing query. In most cases, this is an undesirable situation for the user, because he has to revise his query and send the revised query to the database system in order to get some information from the database. In contrast, flexible query answering systems internally revise failing user queries themselves and by evaluating the revised query return answers to the user that are more informative for the user than just an empty answer. Unfortunately, finding related answers at runtime by consulting the ontology for each query is highly inefficient. Hence it is decisive to apply some preprocessing to the data that enables a more efficient retrieval of related answers to a user's query.

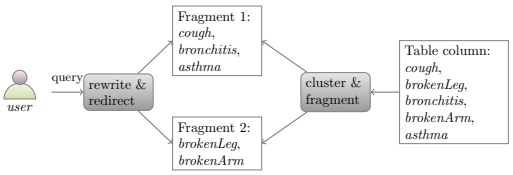


Figure 1: Ontology-based fragmentation

2 Ontology-Based Fragmentation

In previous work [6], a clustering procedure was applied to partition the original tables into fragments based on a *relaxation attribute* chosen for anti-instantiation. Finding these fragments is achieved by grouping (that is, *clustering*) the values of the respective table column (corresponding to the relaxation attribute) and then splitting the table into fragments according to the clusters found.

We assume that each of the clusterings (and hence the corresponding fragmentation) is *complete*: every value in the column is assigned to one cluster and hence every tuple is assigned to one fragment. We also assume that each clustering and each fragmentation are also *non-redundant*: every value is assigned to exactly one cluster and every tuple belongs to exactly one fragment (for one of the relaxation attributes); in other words, the fragments inside one fragmentation do not overlap.

To enable ontology-driven query answering, when a user sends a query to the database, the term (that is, constant) that can be anti-instantiated has to be extracted, the matching cluster has to be identified and then the user query has to be rewritten to return answers covering the entire cluster.

3 Choosing a Clustering Algorithm

The example data set consists of a table (called *ill*) that resembles a medical health record and is based on the set of Medical Subject Headings (MeSH [5]). The table contains as columns an artificial, sequential *tupleid*, a random *patientid*, and a *disease* chosen from the MeSH data set as well as the *concept* identifier of the MeSH entry. We varied the table sizes to allow for different test runs. The smallest table consists of 56,341 rows (one row for each MeSH term), a medium-sized table of 1,802,912 rows and the largest of 14,423,296 rows (obtained by duplicating the original data set 5 times and 8 times, respectively). A clustering is executed on the MeSH data based on the concept identifier (which orders the MeSH terms in a tree); in other words, entries from the same subconcept belong to the same cluster.

The objective of this project is to cluster the MeSH Dataset, and since the number of subcategories is known, a partitioning clustering algorithm (like [1]) is the most appropriate option. K-means [3] and K-medoids are two popular and effective partitioning algorithms which differ from each other based on the construction of the cluster representatives. Partitioning clustering algorithms generate various clusters as output based on the desired number of partitions, considering some criterion. In other words, partitioning clustering simply splits a set of data items into clusters (subsets), so that each data item is not allowed to be in more than one cluster. Clusters are normally detected by iteratively relocating data items between subsets.

4 Experimental Evaluation

Our prototype implementation – the OntQA-Replica system – runs on a distributed SAP HANA installation with 10 database server nodes provided by the Future SOC Lab of Hasso Plattner Institute.

4.1 Clustering with PAL

SAP HANA offers some approaches to move application logic into the database. Utilization of application functions is the most practical one, which with the performance of complicated computations can be improved effectively in the database in comparison with the application sever level. Application Function Libraries (AFL) of SAP HANA include both the Business Function Library (BFL) and The Predictive Analysis Library (PAL). The first one contains functions for common business calculations, while the second one has predictive algorithms, including classification, clustering, association, and more advanced functions. The AFL archive is not part of the HANA appliance, and must be installed separately by the administrator. Since this project deals with clustering of the MeSH dataset, PAL is considered as our target library. In order to confirm the successful installation of PAL and its functions, three public views are supposed to be checked: sys.afl_areas, sys.afl_packages and sys.afl_functions.

In Predictive Analysis Library (PAL), K-means clustering as a method of cluster analysis is implemented using functions which can be called from within SQLScript procedures. In order to use the PAL function, there are two important steps that should be followed: first, a procedure which wraps the PAL function must be created from within SQLScript code and in the second step, it should be called.

Another partitioning clustering method associated to the K-means algorithm is K-medoids. The algorithm is based on medoids calculation as the centroids. In PAL, in order to generate the related procedure, the function name in an AFLLANG procedure generation should be set to 'KMEDOIDS'. The signature table has the same records as mentioned for K-means.

4.2 Clustering with R

R is an open source programming language which is very popular among statisticians and data miners because of its great ability to process and analyze advanced data in terms of volume or complex structure. SAP HANA database requires to be integrated with R, so that embedded codes written in R can be processed in-line as a part of the overall query in the SAP HANA database context. This gives the possibility of making use of R environment for specific statistical functions. SAP does not ship the R environment with the SAP HANA database, so the integration between them needs to be configured by administrator.

The kmeans function from the stats package in R is one the partitioning clustering algorithms. The given data are clustered by the K-means method, which aims to partition the points into k groups such that the sum of squares from points to the assigned cluster centers is minimized. At the minimum, all cluster centers are at the mean of their Voronoi sets (the set of data points which are nearest to the cluster center). PAM (Partitioning Around Medoids [4]) function is a more powerful version of K-means which is present in the clustering package of R. Medoids are similar in concept to means or centroids, but medoids are always members of the data set. The pam-algorithm is based on the search for k representative objects or medoids among the observations of the dataset. After finding a set of k medoids, k clusters are constructed by assigning each observation to the nearest medoid. The goal is to find k representative objects which minimize the sum of the dissimilarities of the observations to their closest representative object. By default, when medoids are not specified, the algorithm first looks for a good initial set of medoids (this is called the build phase). Then it finds a local minimum for the objective function, that

is, a solution such that there is no single switch of an observation with a medoid that will decrease the objective (this is called the swap phase). Compared to the K-means approach in kmeans, the function pam also accepts a dissimilarity matrix, and it is more robust because it minimizes a sum of dissimilarities instead of a sum of squared euclidean distances.

4.3 Experiments

The columns mesh and concept from original table ill are the two main columns on which the clustering will be performed. The columns mesh and concept refer to a medical term and its location in the MeSH tree respectively. Since each medical term has a fixed location (the values of concept are unique), the concept can be taken into account as identifier of each medical term. The most important point in deploying the K-means clustering is that it only works on a numeric data. The K-means algorithm implemented in PAL and R are slightly different from each other. While K-means solely accepts numeric data as input in R, PAL claims to have the ability of processing categorical data as well as numeric data. In the K-means algorithm implemented in PAL, if a column is of categorical type, it will be converted to a binary vector; it means that there should be other columns in integer or double data types in addition to the categorical column.

Since both mesh and concept columns are of categorical data and there is no other numeric column, the K-means algorithm in PAL and R is not able to perform the clustering results. One solution is to map categorical data to a binary matrix. The problem is that creating such a numeric matrix for large-scale data is both inefficient and impractical. Another solution is to convert categorical data to numeric data. Since the data type of column concept is alphanumeric (a combination of alphabetic and numeric characters) i.e A08.637.600.500, the conversion would be achieved in two steps. First the alphabetic character should be mapped to a number, and then the whole string to a number by removing the separator dot (“.”). The K-means clustering algorithm in PAL was executed on 1000 records of the ill table based on the mapped table as the input table and DISTANCE_LEVEL is set to Euclidean distance, MAX_ITERATION to 100 and INIT_TYPE to Patent of selecting the init center (selecting centroids). The K-means function of R was performed on the new data table (1000 records) on the SAP HANA platform. The problem was the unstable results; the contents of the clusters changed by every execution. Therefore PAM function of R is considered which produced stable clustering result. A maximum iteration of 100 and euclidean as the metric was set for the PAM function.

5 Conclusion and Future Work

The results show that PAM clustering function of R is more efficient in comparison with K-means of PAL library in big data clustering operations. The main reason is that the results of K-means are strongly affected by the initial guess of centroids. The four suggested methods (First k observations, Random with replacement, Random without replacement and Patent of selecting the init center) for centroid selection in K-means function of PAL library have weak performance, therefore it led to a relatively unsatisfactory result. A major research question that remains is how to parallelize the clustering step on multiple servers to make the approach fully scalable.

References

[1] B. Delibašić, K. Kirchner, J. Ruhland, M. Jovanović, and M. Vukićević. Reusable components for partitioning clustering algorithms. *Artificial Intelligence Review*, 32(1-4):59–75, 2009.

[2] K. Inoue and L. Wiese. Generalizing conjunctive queries for informative answers. In *Flexible Query Answering Systems*, pages 1–12. Springer, 2011.

[3] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, 1967.

[4] W. S. Sarle. Finding groups in data: An introduction to cluster analysis. *Journal of the American Statistical Association*, 86(415):830–833, 1991.

[5] U.S. National Library of Medicine. Medical subject headings. http://www.nlm.nih.gov/mesh/.

[6] L. Wiese. Clustering-based fragmentation and data replication for flexible query answering in distributed databases. *Journal of Cloud Computing*, 3(1):1–15, 2014.

High-Performance Normalization of Security Log Events

Report for the Project "Security Monitoring and Analytics of HPI FutureSoC Lab (Phase III) in 2015 Fall"

David Jaeger, Andrey Sapegin, Martin Ussath,
Feng Cheng, Christoph Meinel
Hasso Plattner Institute (HPI), University of Potsdam
D-14482 Prof.-Dr.-Helmert-Str. 2-3
{David.Jaeger, Andrey.Sapegin, Martin.Ussath,
Feng.Cheng, Christoph.Meinel}@hpi.de

March 24, 2016

Abstract

Security event logs are an essential tool to detect and monitor attacks in computer networks. However, the number of log events produced in big IT landscapes can grow up to multiple billions per day. Current log management solutions (Security Information and Event Management (SIEM)) cannot even closely normalize such huge amounts of data and therefore disable the tracking of attacks in real-time, which means that the log data remains unusable for attack analysis. As result of our project, we present an approach to fully normalize event logs in high-speed by making use of established high-performance inter-thread messaging in conjunction with a hierarchical knowledge-base of log formats and parallel processing on multiple low-end systems. Using our approach, we are able to process more than 145 000 events/sec on a single machine and can therefore easily handle more than 10 billion events/day, which is enough to handle av-erage and peek loads of log data from big enterprise networks.

1 Event Normalization

In order to understand the presented improvements in event normalization, we first want to give a short overview of the whole process of normalization and where parallelization can be utilized. Figure 1 pictures the workflow of normalization. At the beginning of the workflow a log file or one or multiple servers produce logs. These logs are collected at a central place by the *Log Receiver* component.

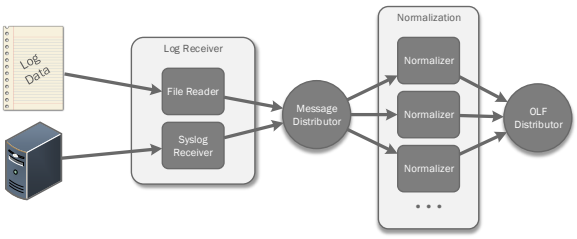


Figure 1: Normalization Process for Event Logs

When the log data is received, it is split up into separate log events, which are then forwarded to the *Normalization* component. Within the *Normalization* component, multiple workers then take over the processing of the logs and normalize it into Object Log Format (OLF). After the normalization, all logs are forwarded for further processing to the *OLF Distributor*.

From the overview, it is obvious that both distributors are a bottleneck in the processing. Usually, the log receiving can be handled by a few workers, because there is no heavy processing to do. The normalization, however, is very processing intensive and needs as many resources as possible and therefore relies on multi-threading. A major

challenge we face in the normalization is to transfer data structures from multiple producers (Log Receiver) to multiple consumers (Normalization). An approach to this challenge is discussed in Section 2.

One *Normalizer* performs normalization with the help of a hierarchical knowledge base. The concept for this kind of normalization has been discussed in our paper [1] and has proven to be highly efficient. The main idea of normalization is to use Named-Group Regular Expressions (NGREs) to match a log line and then use named groups to extract the properties for a newly created normalized OLF event.

2 High-Performance Inter-Thread Communication

Figure 1 shows that normalization is highly dependent on the exchange of event data through the distributors. Taking traditional programming models, this problem of exchange is usually solved with so called blocking queues.

2.1 Blocking Queue Approach

A blocking queue is a thread safe queue implementation that allows a producer thread to put elements into the queue, while one or multiple consumers wait on the queue for incoming events.

In case the queue is empty, the thread blocks and wakes up as soon as the producer puts in a new element. While this model is sufficient for exchanging sporadically incoming elements, it does not perform for a high number of incoming elements, because many CPU time is spent for blocking.

We have used blocking queues for our implementation of normalization, so far. The results for this normalization can be found in the evaluation part of our paper [1]. On average, we could reach around 37 000 events/sec with 8 threads[1] , which is already remarkably high, but can be improved.

2.2 Disruptor Pattern Approach

To increase normalization speed, the so called disruptor pattern[2] can be used. This pattern is based on the exchange of elements through a so called ring buffer and was developed with high-throughput in mind. Because of the parallel access of threads to slots in the buffer, there is no lock required. As soon one thread puts in new data, other threads can read it when they see it.

Because of the promising performance gain, we have implemented the disruptor pattern in our prototypical Real-Time Event Analysis and Monitoring System (REAMS) SIEM to achieve an even higher event throughput. However, to achieve best performance, we had to find the optimal configuration of the disruptor's ring buffer size and the number of producers and consumers. According to the documentation of the disruptor, the ring buffer size should be a power of 2.

Using the machine from the blocking queue implementation of our normalization, we have tested various combinations of ring buffer sizes from 2^7 to 2^{22} slots and normalization threads from 1 to 16 (number of cores on the target machine). By normalizing 10 million Apache web logs, we could find the optimal speed of 64 503 events/sec with the combination of 11 normalization threads and 2^{13} buffer slots.

As a result of our tests, we could already deduce that the disruptor performance is generally higher than the blocking queue performance. Both approaches have almost continuously increasing throughput up to 11 threads. At the point of the highest difference, the disruptor has a 10 755 events/s higher throughput, which estimates to around 20%.

3 Achieving a Lock-Free Implementation

Although there are already good results from the disruptor, we saw a performance drop at 11 threads. While looking for possible reasons for this behavior, we encountered repeated locking behavior in the normalization threads, although the disruptor is supposed to be lock-free.

We were able to pinpoint this locking to one of the programming libraries we used, which were not obviously needing this locking. By replacing and rewriting the affected code sections with lock-free code, we could immediately encounter huge perfor-

[1] Virtual Machine (Debian 7.8, 32GB RAM(dedicated), 16 cores(dedicated)) on VMware ESXi host with 256GB RAM and 8x Intel Xeon X7560 CPUs @ 2.27GHz

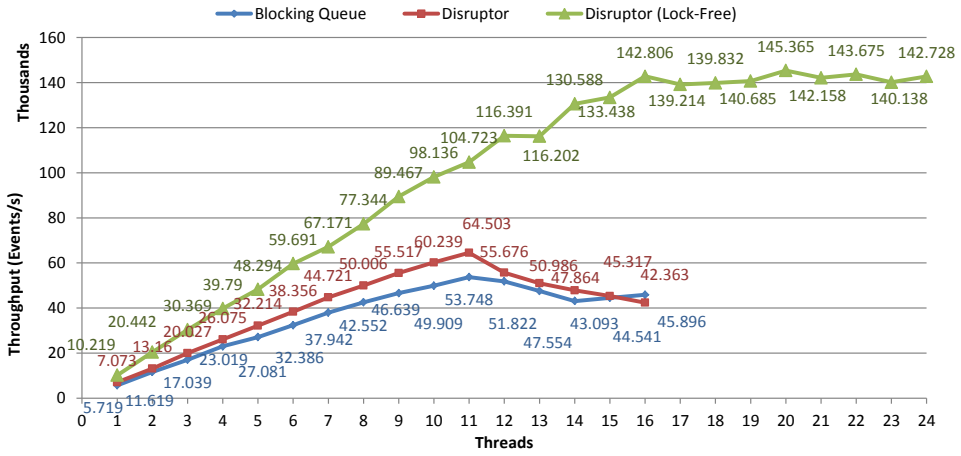


Figure 2: Performance of fully lock-free normalization threads with 16 cores

mance improvements for the normalization threads, as shown in Figure 2.

The lock-free implementation allows us to scale up to a throughput of more than 140 000 events/s with around 16 threads, which is more as double (+125%) the speed as the best throughput of our previous implementation. Additionally, there is now a much more stable growth than before, because now there is no lock contention blocking the processing.

4 Conclusion

In this project, we have shown how multiple billions of events/day, being typical for big enterprise networks, can be normalized in real-time, making an immediate attack analysis, e.g. via an in-memory database, possible. While Gartner categorizes a throughput of more than 25 000 events/s as large setup, this still cannot account for peeks in incoming log events for big enterprises. However, especially such peeks can indicate an malicious activity going on. With our results, we can even handle the normalization of security logs in such occasions.

Our performance comparisons have shown, that a single node can already achieve a normalization throughput of up to 145 000 events/s with a lock-free disruptor implementation.

For future research, we would focus on the fast persistence of the normalized events, so that they are available for further processing in the database, such as for anomaly or misuse detection.

Acknowledgment

We would like to thank HPI FutureSoC[2] lab for providing us with the latest and powerful computing resources, which make the testing and experiments specified in the paper possible.

References

[1] D. Jaeger, A. Azodi, *et al.*, "Normalizing security events with a hierarchical knowledge base," in *Proceedings of the 9th International Conference on Information Security Theory and Practice (WISTP'15)*, vol. 9311, 2015, pp. 238–248. DOI: 10.1007/978-3-319-24018-3.

[2] M. Fowler. (Jul. 2011). The lmax architecture, [Online]. Available: http://martinfowler.com/articles/lmax.html (visited on 09/02/2015).

[3] D. Jaeger, A. Sapegin, *et al.*, "Parallel and distributed normalization of security events for instant attack analysis," in *Proceedings of the 34th IEEE International Performance Computing and Communications Conference (IPCCC'15)*, Dec. 2015.

[2] http://hpi.de/en/research/future-soc-lab.html

Resource Allocation Strategies for Elastic Data Stream Management Systems

Thomas Heinze[1,3], Zbigniew Jerzak[2], Christof Fetzer[3]

[1]SAP SE
Robert-Bosch-Strasse 30/34
69190 Walldorf, Germany
thomas.heinze@sap.com

[2]SAP SE
Rosenthaler Str. 30
10178 Berlin, Germany
zbigniew.jerzak@sap.com

[3] TU Dresden
Noethnitzer Str. 46
01187 Dresden, Germany
christof.fetzer@tu-dresden.de

Abstract

Elastic scaling allows cloud-based data management systems to handle unpredictable load changes by dynamically adding or removing resources. Dynamic resource (de)allocation increases the system utilization and reduces the operational cost. In this proposal we perform a large-scale evaluation of costs and SLAs in elastic data stream management systems. In our evaluation we focus on strategies, which decide where the load is moved.

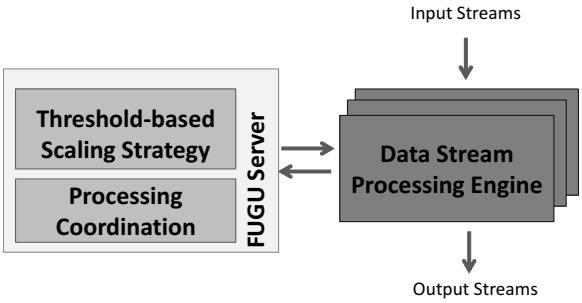


Figure 1: Architecture of FUGU

1 Introduction

Due to a constantly changing workload the utilization of cloud-based systems constantly varies. The utilization of a typical cloud-based system rarely exceeds 30% [11]. The major goal of all providers of cloud-based systems is to maximize their utilization while guaranteeing service level agreements (SLAs) for end users. Maximizing the utilization can be achieved by dynamically allocating and de-allocating resources (hosts). However, the higher the utilization of a given system the more difficult it is for such a system to fulfill user specified SLAs, such as latency and throughput guarantees. This fundamental trade-off is the main motivation driving the research behind the elastic scaling of data management systems.

In our current research we focus on elastic scaling of data stream management systems [6, 8] and publish/subscribe systems [3]. Our concepts have been implemented within a prototype, which uses different scaling as well as optimization techniques in order to achieve the best trade-off between utilization and user-specified SLAs. Our previous research focused on identifying the right scaling policies and determine which operators to move. We like to use the opportunity offered by the HPI Future SOC Lab to study an open problem in our system: the question on which host to place the moved operators. We used simplistic heuristics for this purpose so far [6, 8]. In a next step, we want to evaluate these heuristics in more detail and derive novel algorithms suited best to our use case.

In this report, we summarize the results we achieved in context of the HPI Future SOC Lab Fall 2015. First, we describe the architecture of our prototype in more detail in Section 2. The operator placement problem is introduced in Section 3 and some preliminary results are presented in Section 4. Finally, we describe some conclusions and possible next steps in Section 5.

2 Background

The concepts presented here are implemented as an extension of the elastic data stream management prototype FUGU [6, 7] (see Figure 1). The existing system consists of a centralized management component, which dynamically allocates a varying number of hosts. The manager executes on top of a distributed data stream management engine, which is based on the Borealis semantic [1].

The data stream management system processes continuous queries, which can be modeled as directed acyclic graphs of operators. Our system supports primitive relational algebra operators (selection, projection, join, and aggregation) as well as additional data stream processing specific operators (sequence, source, and sink). Each operator can be executed on an arbitrary host and a query can be partitioned over multiple hosts. The number of hosts is variable and

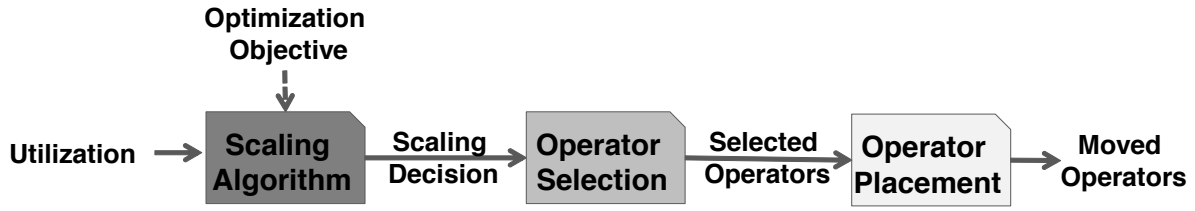


Figure 2: Scaling Strategy of FUGU

dynamically adapted by the management component to changing resource requirements.

The centralized management component serves two major purposes: (1) it derives scaling decisions, including decisions on allocating new hosts or releasing existing hosts, and assigns operators to hosts; and (2) it coordinates the construction of the operator network in the distributed data stream management engine.

The management component constantly receives statistics from all running operators in the system. Based on these measurements and a set of thresholds and parameters, it decides when to scale and where to move operators. Typically, these thresholds and parameters are manually specified by the user. Our system supports the movement of both stateful (join and aggregation) and stateless operators (selection, sink, and source). A state of the art movement protocol [6, 13] ensures an operator moves to the new host without information loss.

2.1 Threshold-based Elastic Scaling

The scaling approach used by the FUGU server is illustrated in Figure 2. A vector of node utilization measurements (CPU, memory, and network consumption) and a vector of operator utilizations are used as input to the *Scaling Algorithm*. The *Scaling Algorithm* derives decisions that mark a host as overloaded or the system as underloaded. The *Operator Selection* algorithm decides which operators to move and the *Operator Placement* algorithm determines where to move these operators.

The default scaling strategy of FUGU is threshold-based, namely, a set of threshold rules are used to define when the system needs to scale. These thresholds mark either the entire system or an individual host as over/underloaded. A threshold rule describes an exceptional condition for the consumption of one major system resource (CPU, network, or memory), which triggers a scaling decision in FUGU. Some examples for these rules include:

1. A host is marked as overloaded if the CPU utilization of the host is above 80% for three seconds.

2. A host is marked as underloaded if the CPU utilization of the host is below 30% for five seconds.

The threshold-based rules need to be used carefully [5]. In particular, the frequent alternating allocation and deallocation of virtual machines, called thrashing, should be prevented. Several steps are taken in FUGU to avoid thrashing. First of all, each threshold needs to be exceeded for a certain number of consecutive measurements before a violation is reported. This number is called the *threshold duration*. In addition, after a threshold violation is reported, no additional scaling actions are done for the corresponding host for a certain time interval called a *grace period* (or cool-down time).

The load in a data stream management system is partitioned among all operator instances running in the system. Therefore, each scaling decision needs to be translated into a set of moved operators. The first problem is to identify which operators to move. This identification is done by the *Operator Selection* algorithm. If the system is marked as underloaded, it selects all operators running on the least loaded hosts. For an overloaded host, the *Operator Selection* algorithm chooses a subset of operators to move in a way, that the summed load remaining on the host is smaller than the given threshold. FUGU models this decision as a *subset sum problem* [9], where the operators on the host are the possible items and the threshold represents the maximum sum. We use a heuristic, which identifies the subset of all operator instances whose accumulated load is smaller than the threshold and no other subset with a larger accumulated load fulfilling this condition exists. All operators selected by this algorithm are kept on the host; the remaining operators are selected for movement.

The selected operators are the input of the *Operator Placement* algorithm, which decides *where* the operators should be moved. An operator can only be moved to a host, if the host has enough remaining CPU, network and memory capacity. The used heuristics can try to fulfill different objectives as discussed in the next section.

3 Operator Placement

The primary task of the operator placement is to assign operators to hosts in a way that the total number of hosts is minimized. Bin packing algorithms [4] are a well known solution to achieve this objective. A bin packing algorithm searches for an assignment of a set

of items to a set of bins. Each item has a weight and each bin has a capacity. The goal of a bin packing algorithm is to assign each item to exactly one bin in a way that (1) the number of bins is minimized and (2) the sum of the weights of all assigned items is smaller than the capacity of the bin. In the context of FUGU, an operator represents an item and its CPU usage its weight. A host is modeled as a bin with its CPU resource as the capacity. In addition, we use network and memory consumption as sub-constraints.
We implemented three well-known bin packing methods to study their performance for our problem:

FirstFit iterates over all available hosts based on the host ID, starting with host 1. An operator is placed on the first found host with enough capacity.

BestFit always studies all available hosts before placing an operator. The operator is placed on the host, which has enough capacity and the largest utilization of all hosts with enough capacity. This approach should minimize the unused capacity on all used hosts.

WorstFit always studies all available hosts before placing an operator. The operator is placed on the host, which has enough capacity and the smallest utilization of all hosts with enough capacity. This approach tries to achieve a balanced load between all used hosts.

These heuristics are well-studied and known for their good performance in terms of minimizing the number of used hosts. However, the problem for our elastic operator placement is slightly different, because the operator placement is executed not only once, but each time an overload or underload is detected. Therefore, the derived decision might be a good solution for the current situation, but can result in some drawbacks for later decisions. In our experiments, we observed, that using the *BestFit* heuristic increases the probability of overloaded hosts, because operators are always moved to hosts with already high load. Similarly, the scale in decision becomes very expensive for the *WorstFit* heuristic as all hosts have comparable load. Therefore, we introduced a novel heuristic, called *Utilization-based FirstFit*, to overcome these problems.

3.1 Utilization-based FirstFit

The major idea behind our novel heuristic is to place the load always on non-critical hosts. These hosts are not closed to get overloaded and are also not likely to be released with the next scale in decision. A characteristic example of such a host is a host with a load close to the medium between lower and upper utilization thresholds. The heuristic sorts all hosts with enough capacity based on how critical they are and starts always with the non-critical hosts first. Afterwards, it places the moved operator on the first host of the list.
We use two metrics to determine how critical a host is: its current utilization and the utilization trend. The utilization is categorized into five classes: very low (below the lower utilization threshold), low, medium, high, very high (above the upper utilization threshold). The three classes low, medium, high are derived by equally partitioning the interval between lower and upper threshold into three partitions. Based on the described heuristic, the hosts are sorted using the following class ordering: Medium, Low, High, VeryLow, VeryHigh. If two or more hosts belong to the same class, we sort them based on the recent utilization trend. The utilization trend describes the observed slope of the utilization in the last ten measurements. The slope is determined using linear regression. Hosts with a negative slope (a decreasing utilization) are a preferred target for moved operators.

4 Preliminary Evaluation

We evaluated the different bin packing heuristics using the hardware provided by the HPI Future SOC lab. Our tests were executed on 10 VM's with 2 cores and 2 GB RAM each. A major effort during our experiments was dedicated to setup our deployment inside the HPI Future SOC lab on these new VM's. The novel setup requires also a set of initial experiments to adjust system and workload parameters.
Afterwards, we run several experiments for five different workloads, two workloads from the energy domain and three based on Twitter data [6, 8]. All experiments where run with the same utilization thresholds, an upper threshold of 0.8 and a lower threshold of 0.3. Each experiment lasted for 60 minutes. We use two major metrics for our evaluation: the monetary cost and the total number of moved operators. We use a pay per use model according to the Amazon EC2 [2], which charges $\$0.135$ per virtual machine per hour. We scaled the reservation time and the prices to a minimum usage time of one minute due to the short experiment duration. The total number of moved operators are used as an indicator for the effects described previously. A wrong placement decision may lead to many additional operator movements in subsequent scaling decisions. For both metrics a smaller value is preferred.
The achieved performance for different bin packing heuristics is presented in Figure 3. The results show for all workloads a huge difference in the number of moved operators for different methods, e.g. for the workload *Twitter Week1* the method *FirstFit* moved up to 120 operators, while *WorstFit* only moves 38 operators. Overall, our novel method *UtilFirstFit* moves in average the smallest number of operators of all studied heuristics. Also the monetary cost varies based on

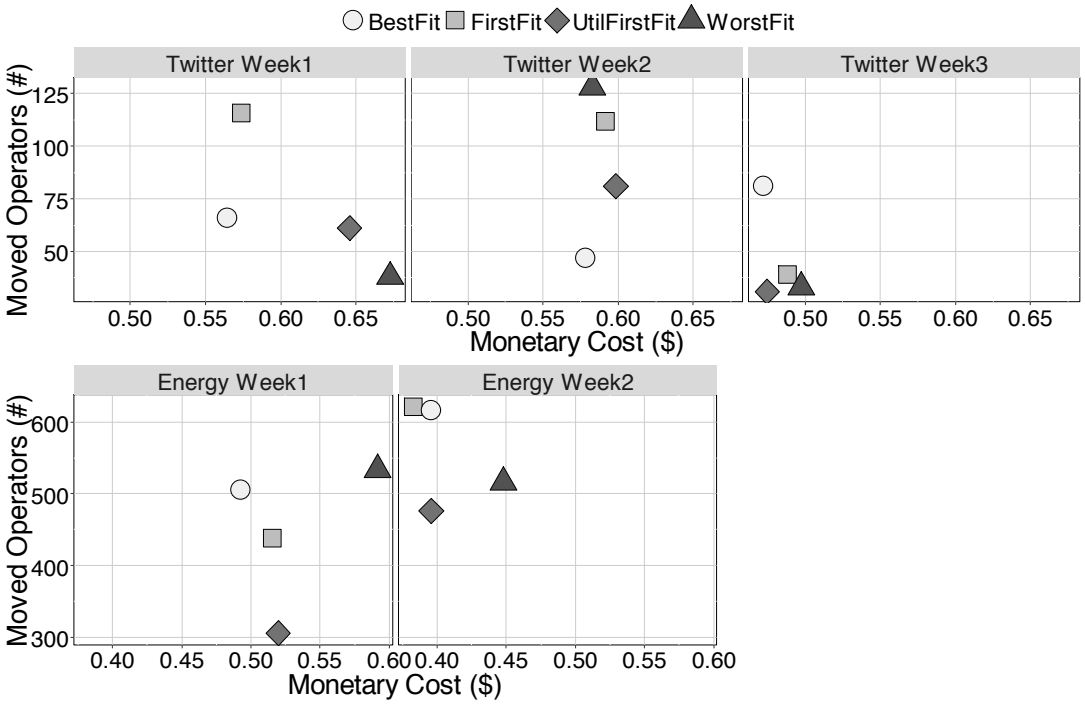


Figure 3: Evaluation Results for Different BinPacking Methods

the used bin packing heuristic, however with a significantly smaller variation. Overall, the smallest monetary cost is measured for the heuristic *BestFit*.
We conclude, that the used bin packing heuristic influences both the achieved monetary cost and number of moved operators. However, the influence of the bin packing heuristic on both metrics is smaller than for the used thresholds [7, 8] and operator selection strategy [6]. The studied heuristics clearly show a trade-off between these two metrics, where no single best heuristic can be identified. We also saw certain potential to improve the results of well established heuristics by a novel heuristic tailored to our problem. However, a more carefully evaluation and improvements of the used approach is required.

5 Conclusion

Elastic scalability is an important property of modern data management systems as it is the key to provide a cost efficient execution. This requirement is especially important for data stream management systems, where the workload varies significantly due to changing data stream rates. In context of the HPI Future SoC Lab Fall 2015 we analyzed the elastic scaling data stream management system, where we focused especially on operator placement algorithms. These algorithms decide on which host to move an operator. We studied different well-established heuristics and compared them with a novel heuristic. We saw some tuning potential based on some early results, but the studied heuristics indicate a clear trade-off between the total number of moved operators and the monetary cost. In the next period of the HPI Future SOC Lab, we like to continue this study. Especially, we like to increase the number of experiments and tune the presented heuristics. In addition, we like to study alternative approaches proposed by other authors in context of similar problems, including min-cut graph partitioning [10], online bin packing [14] or tabu search [12].

References

[1] D. J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. Maskey, A. Rasin, E. Ryvkina *et al.*, "The Design of the Borealis Stream Processing Engine," in *CIDR '05: Proceedings of the Second Biennial Conference on Innovative Data Systems Research*, 2005, pp. 277–289.

[2] Amazon, "Amazon EC2," http://aws.amazon.com/ec2/, accessed November 22th, 2015.

[3] R. Barazzutti, T. Heinze, A. Martin, E. Onica, P. Felber, C. Fetzer, Z. Jerzak, M. Pasin, and E. Rivière, "Elastic Scaling of a High-throughput Content-based Publish/Subscribe Engine," in *ICDCS '14: Proceedings of the 2014 34^{th} IEEE International Conference on Distributed Computing Systems*. IEEE, 2014, pp. 567–576.

[4] E. G. Coffman Jr, M. R. Garey, and D. S. Johnson, "Approximation Algorithms for Bin Pack-

ing: A Survey," in *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996, pp. 46–93.

[5] H. Ghanbari, B. Simmons, M. Litoiu, and G. Iszlai, "Exploring Alternative Approaches to Implement an Elasticity Policy," in *CLOUD '11: Proceedings of the IEEE International Conference on Cloud Computing*. IEEE, 2011, pp. 716–723.

[6] T. Heinze, Z. Jerzak, G. Hackenbroich, and C. Fetzer, "Latency-aware Elastic Scaling for Distributed Data Stream Processing Systems," in *DEBS '14: Proceedings of the* 8^{th} *ACM International Conference on Distributed Event-Based Systems*. ACM, 2014, pp. 13–22.

[7] T. Heinze, V. Pappalardo, Z. Jerzak, and C. Fetzer, "Auto-scaling Techniques for Elastic Data Stream Processing," in *ICDEW '14: Workshops Proceedings of the* 30^{th} *International Conference on Data Engineering Workshops*. IEEE, 2014, pp. 296–302.

[8] T. Heinze, L. Roediger, A. Meister, Y. Ji, Z. Jerzak, and C. Fetzer, "Online Parameter Optimization for Elastic Data Stream Processing," in *SoCC '15: Proceedings of the ACM Symposium on Cloud Computing 2015*. ACM, 2015, pp. 276–287.

[9] S. Martello and P. Toth, "Algorithms for Knapsack Problems," *Surveys in Combinatorial Optimization*, vol. 31, pp. 213–258, 1987.

[10] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of Data Center Networks with Traffic-aware Virtual Machine Placement," in *Proceedings of 2010 IEEE INFOCOM*. IEEE, 2010, pp. 1–9.

[11] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis," in *SoCC '12: Proceedings of the Third ACM Symposium on Cloud Computing*. ACM, 2012, p. 7.

[12] J. Schaffner, T. Januschowski, M. Kercher, T. Kraska, H. Plattner, M. Franklin, and D. Jacobs, " RTP: Robust Tenant Placement for Elastic In-Memory Database Clusters ," in *SIGMOD '11: Proceedings of the SIGMOD International Conference on Management of Data*, 2013, pp. 773–784.

[13] M. A. Shah, J. M. Hellerstein, S. Chandrasekaran, and M. J. Franklin, "Flux: An Adaptive Partitioning Operator for Continuous Query Systems," in *ICDE '03: Proceedings of the* 19^{th} *IEEE International Conference on Data Engineering*. IEEE, 2003, pp. 25–36.

[14] W. Song, Z. Xiao, Q. Chen, and H. Luo, "Adaptive Resource Provisioning for the Cloud using Online Bin Packing," *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2647–2660, 2014.

Fluid-Flow Approximation using ETL Process and SAP HANA Platform

Tadeusz Czachórski
Institute of Informatics
Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
tadek@iitis.pl

Monika Nycz
Institute of Informatics
Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
monika.nycz@polsl.pl

Tomasz Nycz
Institute of Informatics
Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
tomasz.nycz@polsl.pl

Abstract

The paper presents an attempt to translate the fluid-flow approximation algorithm into the ETL process, designed and run using SAP Data Services tool. In particular we investigate possible and efficient methods to easily customize and manage the modelling process. The preliminary results of our research indicated that modelling using ETL process is possible and allows to perform fast data analyzes during model calculations.

Introduction & Project idea

Modern computer networks are constantly subjected to transient queue analysis. The main aim of modelling time-dependent flows and the dynamics of router queues changes is to have a possibility to predict QoS factors, such as packet loss probability and queuing delays. To achieve that we need efficient modelling tools. The ideal program should be able to generate, process and store large amounts of data, that are the results of the numerical calculations. However, the analysis of the changes in vast computer networks, like the Internet, assumes iterative, step-based calculations on large structures that depend on each other, so parallelization capabilities are limited. Therefore, the project aims to explore the possibility of transferring the modelling logic into an ETL process, which is much more customizable and user-friendly for an end user.

The project is an extension of the previous projects entitled "Modelling wide area networks using SAP HANA in-memory database" and "SAP HANA Graph Engine as a network modelling tool".

Fluid-flow approximation model

There are few methods of modelling time-dependent flows, such as: Markov chains, diffusion approximation and fluid-flow approximation. However, the most adequate approach for modelling transient states in wide area networks, including the Internet, is the fluid-flow approximation, [2, 1]. The method uses first-order ordinary linear differential equations for calculations that are solved numerically.

The two basic model equations focus on the changes of the queues in the nodes on the path, eq. (1) and the changes in transmission rates in particular TCP flow, eq. (2). The modification of queue length in one router is defined as the input stream reduced by output stream. In turn, the window grows in the absence of loss on the path, and decreases otherwise.

$$\frac{dq_v(t)}{dt} = \sum_{i=1}^{K} \frac{W_i(t)}{R_i(\boldsymbol{q}(t))} \cdot M_i \cdot (1 - p_v(t)) + - \mathbf{1}(q_v(t) > 0) \cdot C_v \quad (1)$$

$$\frac{dW_i(t)}{dt} = \frac{1}{R_i(\boldsymbol{q}(t))} - \frac{W_i(t)}{2} \cdot \frac{W_i(t-\tau)}{R_i(\boldsymbol{q}(t-\tau))} \cdot \cdot \left(1 - \prod_{j \in V} (1 - P_{ij})\right), \quad (2)$$

Besides the variations of actual window size, the changes in a single flow are dependent on the Round Trip Time, eq. (3), that is the time needed for information about the current network state to propagate through network. RTT values are calculated as the total queue delays in all nodes defined along the connection and the total link propagation delay.

$$R_i(\boldsymbol{q}(t)) = \sum_{j=1}^{K} \frac{q_j(t)}{C_j} + \sum_{j=1}^{K-1} Lp_j \,. \quad (3)$$

The routers additionally have mechanisms preventing overloading their buffers, such as RED, which proactively drop packets when queues exceed certain established thresholds with probability $p_v(t)$:

$$p_v(x_v) = \begin{cases} 0, & 0 \leqslant x_v < t_{min_v} \\ \frac{x_v - t_{min_v}}{t_{max_v} - t_{min_v}} p_{max_v}, & t_{min_v} \leqslant x_v \leqslant t_{max_v} \\ 1, & t_{max_v} < x_v \end{cases}, \quad (4)$$

where x is the weighted average queue length and t_{min}, t_{max} are the thresholds values.
The fluid-flow differential equations are solved numerically. However, if we consider a thousand- and million-node topologies, the calculations generate a large amount of data - for 134023 nodes, 50000 flows and 1000 modelling steps, we obtained more than 50 mln of generated rows for losses, 50 mln for flows and 134 mln for routers. The standard solution is to create a dedicated software structure for storing and analyzing the obtained data. However, we must take into account the two main goals:
1) to mine the knowledge of processes and states of the network in a short period of time;
2) to be able to perform a variety of complex relationship analyzes in the network.
In such cases the use of the dedicated structure leads to the necessity of development of new code, each time the new demand comes. Thus, the more universal solution appears to be the modelling with the use of the database, in particular the ETL processes, that feed the database with generated data.

Methodology & Findings

The studies assumed the implementation of fluid-flow model as ETL job using SAP Data Services and its execution to obtain the numerical calculations. As a result, we analyzed some interesting cases extracted from collected data using SQL queries.
The implementation phase focused on the possibility to implement numerical logic. The fluid-flow algorithm, fig.1, was divided into two parts: initialization (initial step) and calculations (loop over steps, fig.2). Within each part the data were processed. In initialization, the values (such as queue length, congestion window size, drop probability, etc.) were computed in time $t = 0$. In calculations, in turn, the values were computed within the time range $[step\_size; total\_time]$.
Few methods were tested in order to obtain full push-down of the logic into the SAP HANA database. In this paper we select exemplary data flow, within which the flows parameters per single step were computed, fig.3, fig.4.
The research showed that the main benefit of the presented approach are the flexibilities of modification and testing the solutions. Moreover, during the load (calculations) the data successively appear in the database, thus the detection of changes in the network can be performed in the meantime. However, at the

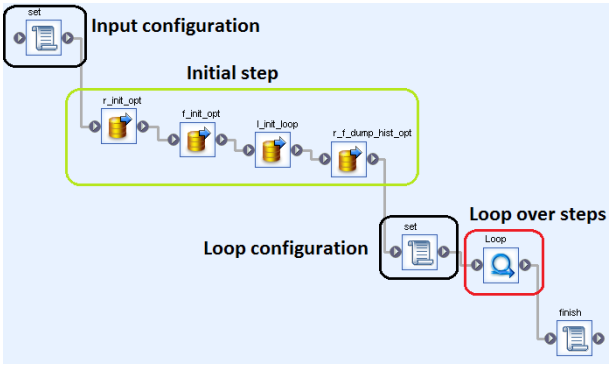


Figure 1: ETL Job View - the main components of the algorithm

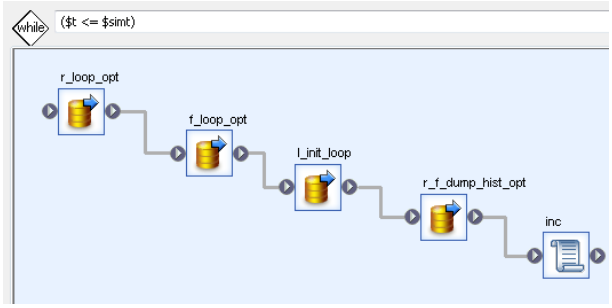


Figure 2: ETL Loop View - the elements of the calculations part

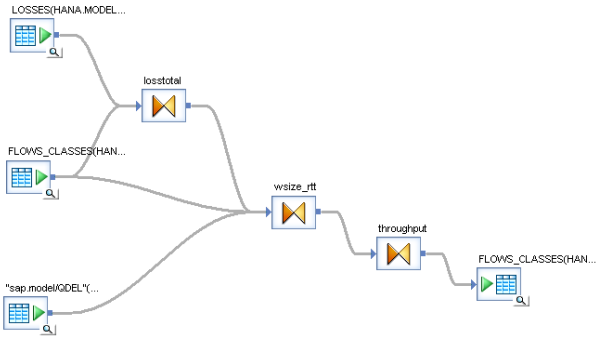


Figure 3: ETL Data Flow View - one of few tested methods of computation of parameters within single step for all flows

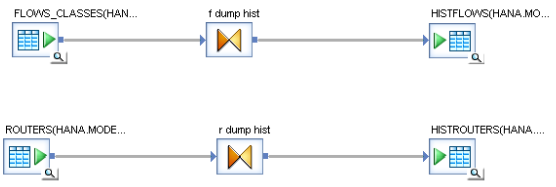


Figure 4: ETL Data Flow View - the method of saving the computed routers and flows parameters at the end of each step

moment the time efficiency of the solution leaves a lot to be desired.
The analysis performed over the data can be very fast (up to few second, depending on the complexity). The possibility to extract knowledge from the data is only limited by the ability to write the query in SQL language. Here we demonstrate the answers to few exemplary questions about network changes and the results.

- The 15 highest percentage queue loads in particular moment in time ($t = 13\ s$)

```
SELECT hr.idrouter,
hr.queue/r.buffer*100 AS load
FROM HistRouters hr
INNER JOIN Routers r
ON r.idrouter = hr.idrouter
WHERE time = 13
ORDER BY load DESC, idrouter
```

Input set records without filters: 134 157 023.
Records processed: 134 023.
Query executed in: ≈ 320 ms.

	IDROUTER	LOAD
1	49,675	87.931091419773181481481481481481481
2	53,890	72.25741631495815
3	26,890	70.337573785835487804878048780487878
4	31,745	70.337573785835487804878048780487878
5	11,271	68.047056745581358585858585858585586
6	49,695	65.070590623553673456790123456790123
7	68,678	63.420218218576079047619047619047619
8	16,175	62.949506915614020833333333333333333
9	16,282	62.949506915614020833333333333333333
10	46,255	62.949506915614020833333333333333333
11	126,933	60.559979985410247826086956521739130
12	127,804	60.559979985410247826086956521739130
13	71,091	59.835904252136616666666666666666667
14	48,309	59.207998707883841666666666666666667
15	46,736	56.778810171094781944444444444444444

- The highest value of RTT time in secs ($t \in [0; 100]$)

```
SELECT MAX(rtt)
FROM HistFlows
```

Input set records: 50 050 000.
Records processed: 1.
Query executed in: ≈ 1.7 ms.

	MAX(RTT)
1	14.688264458915989

- Loss rates in flows in particular time interval ($t \in [50.01; 60]$)

```
SELECT idflow, time, loss
FROM Losses
WHERE time BETWEEN 50.01 AND 60
ORDER BY time, idflow
```

Input set records without filters: 44 559 219.
Records processed: 4 594 719.
Query executed in: ≈ 2.64 s.

	IDFLOW	TIME	LOSS
1	21,381	50.010004423618008	0.004238022645443
2	22,110	50.010009409472012	0.044760234753675
3	21,627	50.010009698796	0.183093816822268
4	39,329	50.010016037576296	0.150178309054563
5	39,896	50.010018346216254	0.004118969052309
6	12,505	50.010034160535752	0.419034971354393
7	43,458	50.010034721669787	0.07153755631423
8	4,870	50.010034738564889	0.063583671587635
9	33,695	50.010036279065603	0.052740148290204
10	23,356	50.01003746675616	0.171650520651167

• • •

- The most frequently congested router (above 50%, $t \in [0; 100]$)

```
SELECT c.idrouter, c.cnt
FROM
(
SELECT hr.idrouter,
COUNT(hr.time) AS cnt
FROM HistRouters hr
INNER JOIN Routers r
ON hr.idrouter = r.idrouter
WHERE hr.queue/r.buffer*100>=50
GROUP BY hr.idrouter
) c
INNER JOIN
(
SELECT MAX(cnt) AS maxcnt
FROM
(SELECT hr.idrouter,
COUNT(hr.time) AS cnt
FROM HistRouters hr
INNER JOIN Routers r
ON hr.idrouter = r.idrouter
WHERE hr.queue/r.buffer*100>=50
GROUP BY hr.idrouter)
) m
ON m.maxcnt = c.cnt
```

Input set records without filters: 134 157 023.
Records processed: 54 457 002.
Query executed in: ≈ 3.66 s.

	IDROUTER	CNT
1	53,890	279

Future SOC Lab resources

During the project we used the HPI Future SOC Lab HP DL980 G7 server having i. a. 4 x Xeon (Nehalem EX) X7560 and 1 TB RAM. The calculations were performed by running the ETL job using SAP Data Services.

Conclusions & Next steps

The use of the conjunction of ETL process and SAP HANA database resulted in flexibility of the solution, especially in logic modifications, and the capabilities of fast data analysis. The research has showed that the fluid-flow equations can be modelled as ETL job, which is much more customizable than standard native code implementation.
As a next step we will focus on further time-based optimization of the ETL algorithms.

References

[1] Y. Liu, F. L. Presti, V. Misra, D. Towsley, and Y. Gu. Fluid models and solutions for large-scale ip networks. *ACM/SigMetrics*, 2003.

[2] V. Misra, W.-B. Gong, and D. Towsley. A fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red. In *Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM 2000)*, pages 151–160, 2000.

Implementation of a Real-Time Usability Improvement Framework for Business Information Systems based on SAP UI5 and SAP HANA

Sharam Dadashnia, Peter Fettke, Peter Loos
Institute for Information Systems (IWi) at the
German Research Center for Artifical Intelligence (DFKI) Campus D3 2, 66123 Saarbrücken
{sharam.dadashnia | peter.fettke | peter.loos}@iwi.dfki.de

Abstract

Usability nowadays plays an important role in the selection process of business information systems. Especially in the context of user-centric development, the usability of such systems gains more and more importance for customers. Therefore, we implemented an integrated framework for the dynamic usability improvement for software users and an analytics dashboard for software developers. The implementation is based on an integrated front-end framework called SAP UI5-Framework and the in-memory database SAP HANA to ensure optimal computational power for resource-intensive calculations.

1 Introduction

The project *Real-time Usability Improvement for Business Information Systems* aims at investigating the dynamic workflow improvement of process-based information systems. The purpose of the investigation is that nowadays the usability of such systems plays an important role in the selection process of supporting software. This is especially true in the context of user-centric development, where the usability of business information systems is a crucial characteristic of differentiation [1]. However, measuring the usability of such systems by automated means as well as their dynamical enhancement has not extensively been studied. The intention is to evaluate an approach, which improves the usability of web-based business information systems in real-time [2]. Different concepts are evaluated, which build on data gathering methods from web analytics to provide logging mechanisms for user interactions at a detailed level and subsequently process this data by means of data analytics and process mining methods. For the evaluation of the existing concepts, there are certain research objectives and challenges, which have to be processed. An important aspect is to determine which data has to be collected from user interactions. Since a large number of data is logged during the usage of business information systems, an adequate hard- and software architecture is very important. In this project, we focused on a general problem, which is related to the correspondence of interaction data mining and process mining techniques in the context of business information systems. In particular, the following research questions have been investigated:

RQ1: Can existing concepts be implemented on an in-memory database architecture?

RQ2: Can SAP HANA-specific functionality help to improve and accelerate the implementation of these concepts?

Against this background, the remainder of this report is structured as follows: Section 2 describes the research approach that our findings are based on. Next, section 3 shortly elaborates on the real-time usability improvement framework, which is based on the SAP HANA in memory database, before section 4 reports the actually prototype implementation and section 5 concludes the report and gives an outlook on follow-up projects.

2 Research Approach

The research described in this article is based on the concept of architectural prototyping originating from software architecture development. An architectural prototype in that regard represents a learning and communication vehicle for the differentiation of styles, features and patterns of a system under development and helps to explore and evaluate the best alternative in the development process [3]. The main objective of the approach relates to problems regarding the application of adequate process mining techniques with respect to calculated metrics and data from the

field of usability. The problem is faced in an iterative manner: by using a repetitive cycle containing a feedback loop, techniques and corresponding parameters are incrementally refined. Figure 1 visualizes the employed four-step research approach with the possibility of multiple iterations in phases two and three, which are executed on the IT basis infrastructure provided by the HPI Future SOC Lab consisting of a SAP HANA In-Memory Database (24 cores, 250 GB of RAM). The software prototype builds on the HANA XS application base.

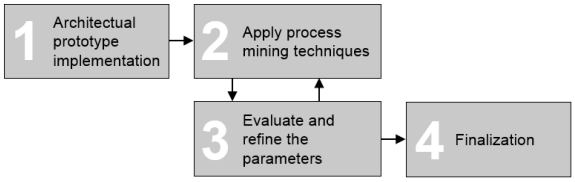


Figure 1: AP research approach

3 Real-Time Usability Improvement Framework

As mentioned before, within this project we implemented certain fragments of our already existing real-time usability framework [2] on the provided hard- and software architecture to ensure that the initial requirements of the framework are considered in the implementation phase.

The implementation of the framework is generally divided in three different phases. The phases are integrated in a comprehensive framework and executed sequentially, as shown in Figure 2.

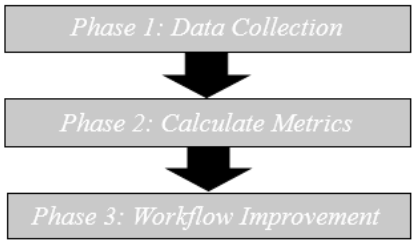


Figure 2: Three Phases of Dynamical Improvement

They provide a summary of the individual steps of the real-time usability improvement framework. In the following, we briefly summarize the different phases and outline a short application scenario for the current implementation stage of the framework.

In *Phase 1*, user interaction data is collected by an appropriate data gathering service based on java script methods which are executed on the respective client. The collected data is directly stored in an in-memory database to provide for instant analyses. The gathered data is stored because we want to provide the necessary information for classical process mining techniques like *caseID*, *eventID*, *timestamp*, *activity* in real-time. The developed gathering methods are directly adaptable to every SAP UI5 application. In this context, we further need some specific data for the web analytics metrics of SAP UI5 application. Therefore, there we also collect *targetId*, *targetView*, *routingArguments* etc. This metrics can be used in the context of software development, to further improve future development cycles. To present the data in an adequate way, we developed a dashboard for developers to support the presentation of the metrics in the first step. The dashboard is called *sAnalytics Explorer* (see Figure 4) and displays results of the web analytics and process mining analyses conducted in the next phase.

Phase 2 describes metrics like the total number of sessions that have been executed with a business information system, or the average duration of the sessions. Within this project and for the purpose of testing, we gathered the data and calculated the metrics for the usage of the application dashboard itself. A selection of the calculated web analytics metrics is shown in Figure 3 on the left side in the section “General Overview” [4]. The dashboard also shows multiple filtering options for selecting the data. On the left sidebar, there is an application chooser. This way, it is possible for developers to track multiple applications in real-time. Furthermore, there are some options in the initial dashboard to discover the log data which are logged on every single click of the user. Besides the distribution of sessions per day, we also have the possibility to see which operating system or device people used to process tasks in the application. Furthermore, there are certain other calculations regarding process mining that are already implemented in the prototype. For

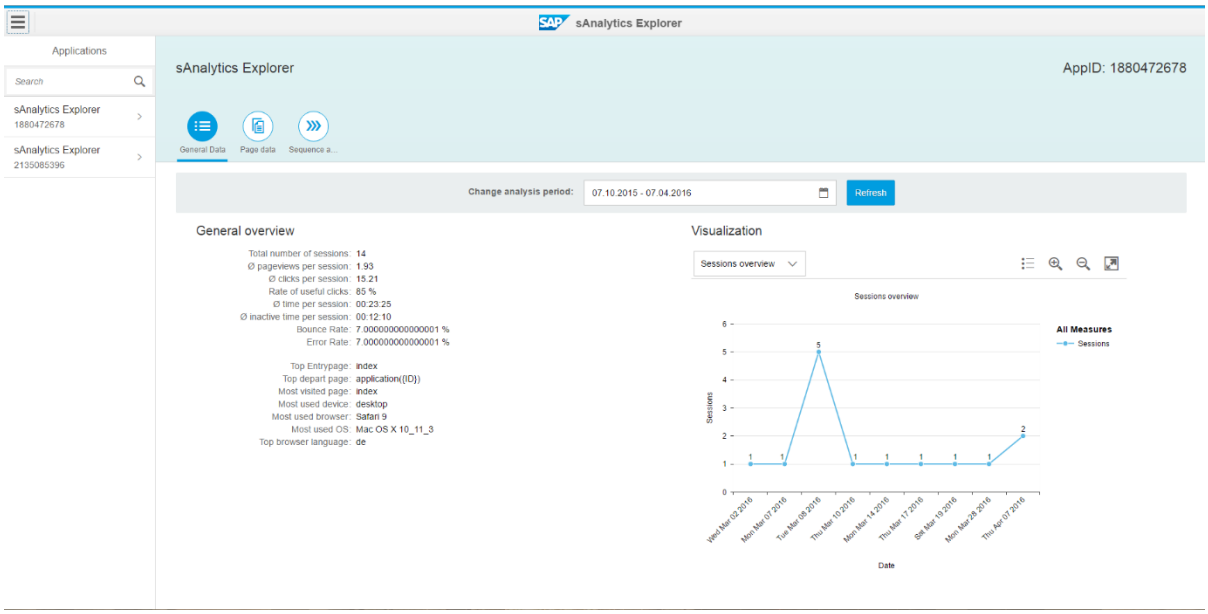


Figure 3: Entry screen of the sAnalytics explorer

process mining, it is an essential step to preprocess the log data as a first step. This is necessary for certain aspects like complexity reduction and process discovery [5]. For this reason, we preliminary implemented certain clustering approaches, namely methods for sequence clustering. These methods are used to combine similar sequences within a data log to ensure an understandable process model before applying process discovery in a subsequent step [5]. The implementation of a visualization for clustering-related task is shown in Figure 4 below.

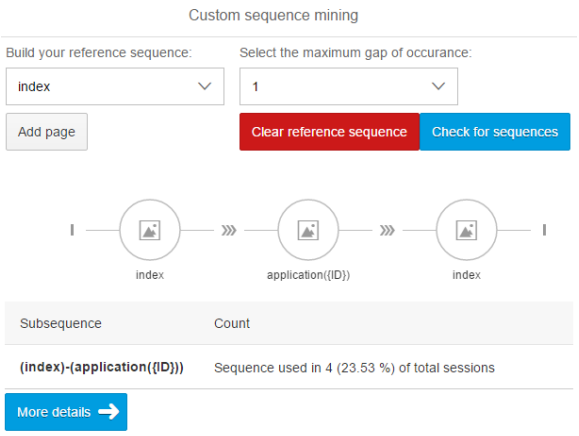


Figure 4: Custom sequence mining

In the upper half of the figure, we implemented an editor, to define custom sequences, which are comparable to certain steps in the application. Those sequences can then be checked against the collected log data. This makes it possible to see which processes in the application are really executed and which sessions they belong to.

In *Phase 3*, the actually workflow improvements based on the calculated metrics are determined. The implementation of this phase is still subject to ongoing research and, thus, not part of the prototype yet. For the implementation, there are also some refinements for phases 1 and 2 necessary in order to provide a solid base for the calculations and real-time improvement techniques [2]. Currently existing process mining methods need to be extended to provide, for instance, prediction possibilities for next process steps etc.

4 Implementation

To implement the prototype, we used certain SAP HANA-specific functionalities like stored procedures for data processing as well as the mining of sequences. Since the standard SAP HANA PAL library does not provide the necessary functionality for the calculation, we used the SAP HANA interface to integrate the statistical tool R [6]. The R engine therefore had to be installed on the same server infrastructure provided by the HPI Future SOC Lab to enable calculations based on stored procedures. Furthermore, the R calculation engine was extended by a specialized library called TraMineR [7]. This extension is especially suitable for simple sequence analysis.

5 Conclusion and Outlook

The project *Realtime Usability Improvement for Business Information Systems* further extends fundamental work that has already been conducted in [2]. Within this project period, we implement parts of the already developed concepts. Regarding Figure 2 the current state of the implementation is that we finished phase 1 and almost phase 2. Phase 3 is planned to be implemented in the following project period. Besides the extension of the analytical dashboard, we also will develop a tool to support the project management process, a so called staffing tool for internal and external employees. This tool constitutes the base for intended evaluations in the futures.

The implementation described in this article showed that the SAP HANA XS platform provides an adequate infrastructure for the conceptual implementations. Furthermore, SAP HANA specific functionalities like stored procedures in connection with external libraries largely facilitated the development process.

Acknowledgement

The provided high performance IT infrastructure from the HPI allowed the investigation of concrete problem fields in information systems research. The authors thank the HPI Future SOC Lab for the chance of using these resources and appreciate a continuation of the project. The basic concepts were developed in context of the project "Echtzeit Usability Verbesserung auf Basis von Mining-Technologien unter Verwendung von In-Memory Computing", which is funded by the Bundesministerium für Bildung und Forschung BMBF (Software Campus).

References

[1] Lambeck, C., Muller, R., Fohrholz, C., & Leyh, C. (2014, January). (Re-) Evaluating User Interface Aspects in ERP Systems - An Empirical User Study. In System Sciences (HICSS), 2014 47th Hawaii International Conference on (pp. 396-405). IEEE.

[2] Dadashnia S.; Niesen T.; Fettke P.; Loos P.: Towards a Real-time Usability Improvement Framework based on Process Mining and Big Data for Business Information Systems, 2016 In: Tagungsband Multikonferenz Wirtschaftsinformatik. Multikonferenz Wirtschaftsinformatik (MKWI-16), March 9-11, Ilmenau, Germany, 2016.

[3] Bardram, J. E.; Christensen, H. B.; Hansen, K. M.: Architectural prototyping: An approach for grounding architectural design and learning. In: Software Architecture, 2004. WICSA 2004. Proceedings. Fourth Working IEEE/IFIP Conference on (pp. 15-24). IEEE.

[4] Peterson, E. T.; Web analytics demystified A Marketer's Guide to Understanding How Your Web Site Affects Your Business, 2004. Celilo Group Media and CafePress. Celilo Group Media. Retrieved.

[5] T. Thaler, S.F. Ternis, P. Fettke, and P. Loos. A comparative analysis of process instance cluster techniques. In Proceedings of the 12th International Conference on Wirtschaftsinformatik. Internationale Tagung Wirtschaftsinformatik (WI-15), March 3-5, Osnabrück, Germany. University Osnabrück, 3 2015.

[6] The R-Project. Link: https://www.r-project.org/, 2016.

[7] TraMineR. http://traminer.unige.ch/, 2016.

Protecting minors on social media with early identity deception detection

Estée van der Walt
Department of Computer Science
Security & Data Science Research Group
University of Pretoria, South Africa
estee.vanderwalt@gmail.com

J.H.P. Eloff
Department of Computer Science
Security & Data Science Research Group
University of Pretoria, South Africa
eloff@cs.up.ac.za

Abstract

Deception has always been a problem in a society with fierce competition to be the best and reach the top or purely for malicious intent. It is difficult to differentiate truth from lies. This even more so on social media platforms where a recipient of information is to rely on words, images and video only to make final judgements on a person's authenticity about themselves and their actions. The research at hand proposes to classify social media deception and build an identity deception indicator to protect the innocent; especially minors with little experience in applying intuition on whether they are being lied to.

1 Project idea

To evaluate deception in social media it is important to first understand what it is. Deception is defined as "The action of deceiving someone" and deceit defined as "The action or practice of deceiving someone by concealing or misrepresenting the truth" [1].

Even after many years it remains difficult to detect deception. Humans are still preferred to use intuition and facts above technology. An example of this is the polygraph test results which are not admissible as evidence in a court trail [2]. A jury is still ultimately entrusted with a ruling on a person's innocence at the end. Research studies have however found that humans themselves are also not too good at detecting deception and are a mere 4% better than having picked a result by chance [2] [3].

Deception is just as prevalent in social media platforms than in any other facet of daily life [3]. With social media, deception has become global and can now reach anyone online. It is also hard to distinguish whether someone is telling the truth without physical contact to that person. In the past a person's body language or voice pitch, for example, could have given away their intention to deceive.

Within social media many threats exist, for example: Social spying, catfishing, online grooming, sexting, social 419 scams, email spamming, online blackmailing and cyber bullying [4] [5] [6]. Many of these threats have an element of deception. Deception can have severe consequences; even online. In 2014 a 14-year-old boy was groomed via online gaming platforms and lured to a house where he was eventually murdered [7].

It is thus important to find means of addressing the threats brought about by deception. We are particularly interested in preventing deception and protecting minors with our research case study although education has also been noted as another means [8].

Existing methods of prevention are either not predictive or do not include the heterogeneous attributes of social media data. With non-predictive we mean that the technology is built to exclude deceivers rather than predicting any malicious activity whilst happening. There will always be new ways to deceive and we believe it will be impossible to keep up with new prevention techniques which will not disable 'good' people from using the network as intended.

With this research project we propose to build an identity deception indicator as early warning to authorities of potential deception on a social media platform. The research project has been divided into various processes discussed in more detail during previous research papers. The focus of this phase of the research was to add more data to the initial sample big data social media dataset. This specific process is highlighted in green in figure 1.

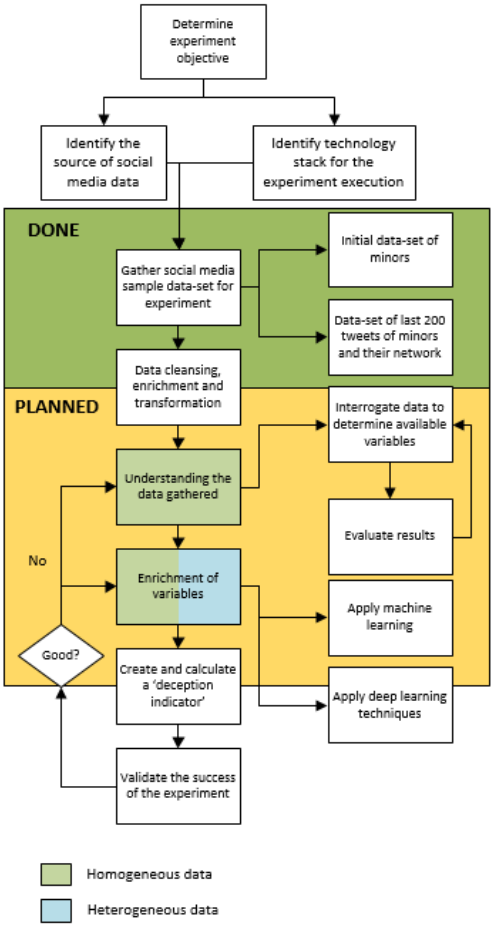


Figure 1: The project process diagram

1.1 Main deliverables

The main deliverables of 2015 were:

- To gather a big dataset for the experiment consisting of minors.
- To add more data to the current identified accounts as well as their friends and followers.
- To automate the collection process to run in a background process for other research to continue
- To clean, enrich and transform the data.
- To understand the data through variable inspection.
- To explore machine learning algorithms for enrichment and addition of more variables to the research at hand.

2 Use of HPI Future SOC Lab resources

To summarize the following resources were used for the research at the HPI Future SOC lab:

- Twitter: The Twitter4j Java API was used to dump the data needed for the experiment in a big data repository.
- Hortonworks Hadoop 2.3: For the purposes of this experiment HDP Hadoop runs on an Ubuntu Linux virtual machine hosted in "The HPI Future SOC"- research lab in Potsdam, Germany. This machine contains 4TBs of storage, 8GB RAM, 4 x Intel Xeon CPU E5-2620 @2GHz and 2 cores per CPU. Hadoop is well known for handling heterogeneous data in a low-cost distributed environment, which is a requirement for the experiment at hand.

  Flume: Flume is used as one of the services offered in Hadoop to stream initial Twitter data into Hadoop and also into SAP HANA.

  Ambari: For administration of the Hadoop instance and starting/stopping the services like Flume.

  Note that we have upgraded our Hadoop instance from version 2.2 to a stable version of 2.3 to enable the use of Spark potentially in the next phase of the project.
- Java: Java is used to enrich the Twitter stream with additional information required for the experiment at hand and automate the data gathering process.
- SAP HANA: A SAP HANA instance is used which is hosted in "The HPI Future SOC"- research lab in Potsdam, Germany on a SUSE Linux operating system. The machine contains 4TBs of storage, 2TB of RAM (1.4TB effective) and 32CPUs / 100 cores. The in-memory high-performance processing capabilities of SAP HANA enables almost instantaneous results for analytics.

  The XS Engine from SAP HANA is used to accept streamed Tweets and populate the appropriate database tables.

  Note that we moved to a SAP HANA instance with more dedicated memory to handle to queries on the large datasets more effectively.
- Machine learning APIs: Various tools are considered to perform classification, analysis and apply deep learning techniques on the data. These include the PAL library from SAP HANA, SciPy libraries in Python, Spark Mlib on Hadoop and the Hadoop Mahout service. We have dropped Graphlab from the set as Spark seems to be more

superior.

- Visualization of the results will be done using the libraries of Python or visualization already in SAP HANA and not HTML as originally planned.

The following ancillary tools were used as part of the experiment:

- For connection to the FSOC lab we used the OpenVPN GUI as suggested by the lab.
- For connecting and configuration of the Linux VM instance we used Putty and WinSCP
- For connecting to the SAP HANA instance we used SAP HANA Studio (Eclipse) 1.80.3

3 Findings in the Spring 2015 semester

The purpose of this phase of the research project was to increase the size of our existing dataset. The initial dataset was a proof of concept which served us well during the previous semester.

We enhanced the Java application pulling data from Twitter into SAP HANA to

- Use 10 Twitter accounts running in separate threads in parallel
- Comply to the rate restrictions employed by Twitter
- Sleep threads when rate limits have been reached
- Appropriate logging of the process for audit trail purposes

We are happy that we are now able to pull in the region of 2 million tweets per hour unaided into SAP HANA.

To this point we have been able to accumulate over 1 million original tweets containing words on 'school' and 'homework'. From this set we extended the dataset with additional tweets of the person, their friends and followers to over 1billion tweets to data as shown in figure 2. This dataset is in the region of about 800GB. We believe that the automation of this process will now enable us to reach our initial goal of 2-4TB of data.

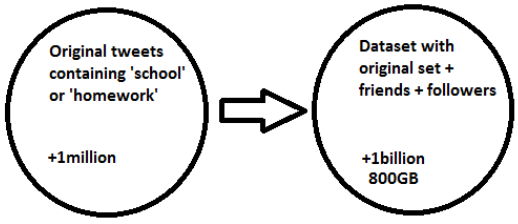


Figure 2: Twitter dataset

The SAP HANA instance, virtual machine and storage was provided by the HPI FSOC research lab and the following is worth mentioning:

- There were no issues in connection.
- The lab was always responsive and helpful in handling any queries.
- The environment is very powerful and more than enough resources are available which makes the HPI FSOC research lab facilities ideal for the experiment at hand

Overall we found that the environment and its power enabled the collection of a big dataset without issue. The support of the HPI FSOC research lab is appreciated.

4 Next steps for 2016

The next steps in the project is to continue with the investigation and analysis of the variables available in the big dataset.

Three datasets have been created for initial investigations:

- 63,226,778 with 29,517 accounts (top3200 tweets -> 38GB)
- 21,446,745 with 29,517 accounts (top1000 tweets -> 12GB)
- 4,764,733 with 6,846 accounts (top1000 tweets pulled March only -> 3GB)

We aim to enhance the dataset with potential data from other social media sites like Facebook and LinkedIn. We strive to have an initial identity deception indicator at the end of the next semester.

The deliverables for this phase are:

- To clean to data based on initial findings from previous data interrogation
- The enhance the dataset with data from other social media sites
- To add more variables for experimentation
- To apply various different machines learning techniques in both SAP HANA and Hadoop
- To evaluate the results from these techniques and identify enriched variables
- To experiment with methods of identifying useful variables and weight their importance
- To produce an initial identity deception indicator per online persona

References

[1] O. Online, "The English Oxford Dictionary," Third Edition, March 2012 ed: Oxford University Press, 2012.

[2] C. F. Bond and B. M. DePaulo, "Accuracy of deception judgments," Personality and social psychology Review, vol. 10, pp. 214-234, 2006.

[3] V. Hauch, I. Blandón-Gitlin, J. Masip, and S. L. Sporer, "Are computers effective lie detectors? A meta-analysis of linguistic cues to deception," Personality and Social Psychology Review, vol. 19, pp. 307-342, 2015.

[4] B. Jäger and P. Leitner, "Innovative Approaches and Solutions to Understand, Identify and Tackle Social Media Crime," Pacific Asia Journal of the Association for Information Systems, vol. 7, 2015.

[5] K. D'Costa, "Catfishing: The Truth About Deception Online," Scientific American, 25 April 2014 2014.

[6] D. M. Cook, "Birds of a Feather Deceive Together: The Chicanery of Multiplied Metadata," Journal of Information Warfare, vol. 13, pp. 85-96, 2014.

[7] A. Moore, "I couldn’t save my child from being killed by an online predator " in The guardian, ed, 2016.

[8] E. Kritzinger, "Enhancing cyber safety awareness among school children in South Africa through gaming," in Science and Information Conference (SAI), 2015, 2015, pp. 1243-1248.

Implementation strategies for policy-aware federated cloud scenarios

Max Plauth, Felix Eberhardt, Frank Feinbube and Andreas Polze
Hasso Plattner Institute for Software Systems Engineering
P.O. Box 90 04 60
14440 Potsdam, Germany
{firstname.lastname}@hpi.de

Abstract

To evaluate the applicability of privacy policy language concepts, we present an exemplary application scenario that involves the provisioning of Hyrise in-memory database instances in an OpenStack*-based cloud environment. For both Hyrise and* OpenStack*, we point out implementation approaches for integrating certain policy attributes into the existing components. However, local testing is merely adequate in order to derive reliable statements about the correct behavior of policy enforcement mechanisms in distributed, cloud-based setups. Using nested virtualization, we address this deficiency by providing a testbed that resembles the basic properties of a production-grade environment.*

1 Introduction

In this report, we discuss the technical implications of employing policy language concepts discussed in [3] by example of the use case scenario illustrated in Figure 1. The scenario includes numerous users, where each user requests an instance of the *Hyrise-R* in-memory database in a *Platform as a Service* (PaaS) like manner. However, users impose certain requirements regarding attributes ranging from the coarse-grained properties such as data center location to fine-grained requirements like database configuration parameters. The *policy decision point* (PDP) acts as the main entry point for users requests. While Figure 4 depicts the *PDP* as a centralized component, its actual implementation strategy might vary. Based on the policies specified by a user, the *PDP* routes requests through a series of *policy enforcement points* (PEP) in order to comply with the respective policies. With policy language concepts at hand, users can impose requirements on service providers by annotating their requests accordingly. On the coarse level, requirements such as geolocation or *Quality-of-Service* (QoS) might be expressed, whereas the fine-grained level can be used to specify application specific demands like availability properties or user rights.

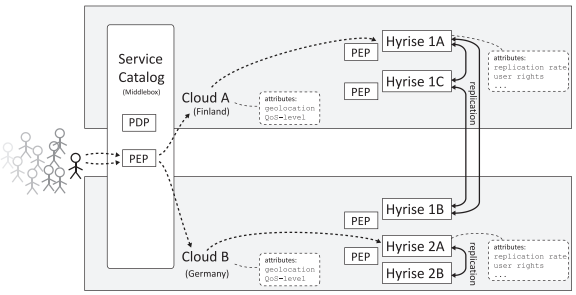


Figure 1: Use case scenario: Users request instances of the *Hyrise-R* in-memory database and annotate their requests with certain policy demands. The *policy decision point* (PDP) acts as the initial entry point and routes requests through a series of *policy enforcement points* (PEP) to process the requests accordingly.

The scenario demonstrates that multiple components have to cooperate in order to consider policy requirements on all levels. In Section 2, we discuss the technical facilities within *Hyrise-R* to achieve different levels of availability. To illustrate this, we provide different *QoS*-levels of the database service and leverage internal mechanisms of the database (k-safety, replication) to satisfy the associated availability policies.
In Section 3, we consider requirements for certain availability policies as well as restrictions on the geographic locality of the services provided in the context of *OpenStack*. Furthermore, we propose a framework that allows us to evaluate the dependability of policy enforcement mechanisms. Using nested virtualization, a federated *OpenStack* setup comprising multiple regions provides the means for performing fault-injection experiments and performance evaluations in a repeatable environment.

2 Policy Implementation in Hyrise-R

This section discusses the implementation of policy language concepts in the scope of the database cluster *Hyrise-R*. It starts with an introduction of the in-memory database *Hyrise* and describes its main-delta

architecture and flexible table layout. We have implemented a scale-out extension of *Hyrise*, called *Hyrise-R*. After giving an overview of the architecture and key concepts, we discuss policy features that are supported by the design of Hyrise-R.

2.1 In-Memory Database Hyrise

Hyrise is an in-memory research database, implementing a main delta architecture like SAP HANA [13]. Tuples in the main partition are stored dictionary compressed with a sorted dictionary to support efficient vector operations and optimized range queries. New tuples are inserted in the write-optimized delta partition with an unsorted dictionary as trade-off for better write and reasonable read performance. The periodic merge process moves tuples from the delta to the main partition [9]. *Hyrise* supports a flexible hybrid table layout to store attributes corresponding to their access patterns [7, 6]. A columnar arrangement is well-suited for attributes which are often accessed sequentially, e.g., via scans, joins, or aggregations. Attributes accessed in OLTP-style queries, e.g., projections of few tuples, can be stored in a row-wise manner. *Hyrise* exploits an insert only approach and multi-version concurrency control with snapshot isolation as default isolation level [14].

2.2 Database Cluster Extension Hyrise-R

We extended *Hyrise* with the capability to form a database cluster. The extension is called *Hyrise-R* [15] and the implemented distribution approach can be classified as lazy master replication [5]. A query dispatcher is the user's access point for submitting database requests and propagates the queries to appropriate cluster nodes. A single database instance, called master node, is responsible for transaction handling. The master node sends its log messages, describing the physical data changes, to the other cluster instances, called replicas. The replica nodes update the data with the log information to keep in sync with the master node. To detect node failures, the master node sends heartbeat requests to the replicas.

2.3 Policy Features in Hyrise-R

Our goal is to employ policy language concepts in *Hyrise-R*. The user will not only be able to store and query data in *Hyrise-R* but also to describe policy properties. However, replication, i.e., storing the same data on each node of the database cluster, does not support all policy features. Given a *Hyrise-R* cluster spread over multiple instances, the user can only store the data on all or none of them. This subsection covers selected policy properties, supported by the design of *Hyrise-R*.

We implemented *Hyrise-R* for read scalability and availability. The number of database cluster nodes can be increased to scale the read throughput. The dispatcher will distribute incoming reads among all cluster nodes. Besides scalability, a higher number of *Hyrise* instances in the cluster increases availability. We will implement K-safety for *Hyrise*. K reflects the number of replicas in the cluster. These replicas can take over the role of the master node in case of a node failure. This requires a failover mechanism and an approach replicating data changes before transactions commit.

The geolocation of cluster instance, i.e., the identification of the real-world geographic location of the computer running the database, is a further policy property a user or *PDP* may want to control in database cloud scenarios. However, the desired policy properties may conflict. On the one hand legal requirements may forbid distributing data and storing it in specific countries. On the other hand companies may want to store their data in different clouds or geolocations to increase availability or decrease latencies. The dispatcher can propagate database requests to instances located close to the user to reduce response latency.

Policy languages can describe access control and rights management. We distinguish database users and their privileges. Privileges can be classified into object and system priviledges. Object priviledges specify end users' rights on database objects, i.e., tables, indices and procedures. They describe for example which SQL operations, e.g., select, insert, update, delete, create, alter, drop, the user is allowed to execute and grant to others and whether he can debug database operations. System priviledges concern the administration of the database, e.g., logging, backups, user management.

3 OpenStack cloud federations

As cloud computing becomes more and more popular, there is an increasing number of implementations to offer various cloud service models like infrastructure as a service (IaaS), platform as a service (PaaS) or software as a service (SaaS). While many companies offer commercial solutions like the Amazon Elastic Compute Cloud (EC2) or HP Helion, there are also open source alternatives that can be freely installed and configured to meet the needs of one's projects with respect to the underlying hardware available.

The *OpenStack* project offers a cloud software stack which allows for offering infrastructure as a service, almost independent of the underlying hardware setup. The project itself can be seen as a collection of services that can be configured according to the specifications of the planned use cases. Central components that *OpenStack* is comprised of are networking, virtualization and storage services. Furthermore, it is possible to add further components to an *OpenStack* installation, e.g., services that handle billing or allow for object storage in the cloud.

When offering a service in cloud computing, it is crucial that customers can rely on service properties such as security mechanisms and dependability. A service should be highly available, meaning that the system should be continuously operational without failing. Furthermore, all policies that the service provider and the consumer agreed upon have to be adhered to. Therefore, our main goal is to integrate a basic set of policy attributes into the *OpenStack* ecosystem. To do this in a controlled environment, we chose to manually set up a clean single-node *OpenStack* environment (i.e., none provided by a third party like HP Helion) on which we would be able to run the specific analyses. In our recent efforts [4], we turned this manual installation into an automated process in order to simplify and speed up the process of setting up a working *OpenStack* test environment and making the resulting analyses repeatable. Since no *OpenStack* installation is exactly the same, the repeatability of the results of such analyses is not an easy feat. We tackle this issue by making the test environment for the experiments completely virtual. Thus, we circumvent tedious hardware setup and hardware errors that disturb the experiments.

Within a single *OpenStack* instance, mechanisms such as service replication can be used to ensure that certain availability requirements are met. In setups where multiple instances of *OpenStack* are interconnected in order to form a federated cloud, improved availability properties can be implemented. In single region setups, the coarse grained choice of the region is sufficient in order to adhere to geolocation policies. For federated setups that span across multiple countries however, complying with geolocation attributes requires more fine-grained mechanisms that enable individual requests to be processed in the proper location.

At its current state, our virtualized testbed [4] can automatically create a setup consisting of a single *OpenStack* instance. However, our goal is to study federated *OpenStack* environments, that are comprised of multiple *OpenStack* instances. An overview of the available methods for creating such federated setups based on *OpenStack* are presented in Section 3.1. In this Section, we also discuss potential approaches for integrating certain aspects of policy language constructs into the *OpenStack* ecosystem. Finally, Section 3.2 documents our ongoing efforts in extending our single-instance setup to a federated environment. In this Section, we also propose the application of fault injection mechanisms in order to validate the correct behavior of the policy enforcement mechanisms we intend to provide for the *OpenStack* ecosystem. In addition to test cases that investigate the adherence to policy attributes, the virtualized testbed provides the means for evaluating non functional parameters such as performance metrics.

3.1 Review of Federation Mechanisms

At the time of writing, the *OpenStack Architecture Design Guide* [12] differentiates between two approaches for interconnecting multiple *OpenStack* instances in order to form a federated setup. In this section, the architecture of each approach is presented and opportunities for integrating policy enforcement mechanisms are discussed.

3.1.1 Cloud Management Platforms

The approach based on *Cloud Management Platforms* (CMPs) assumes mostly unaltered *OpenStack* instances, which are coordinated by a broker-like entity, the so-called *Cloud Management Platform* (see Figure 2). The main advantage of *CMPs* is that they require very few to no alterations of existing *OpenStack* instances. This property can be traced back to one of the main goals of *CMPs*, which is to abstract from the underlying cloud platform in order to support hybrid cloud setups and cloud bursting scenarios. While this abstraction may be beneficial for cloud-bursting scenarios in hybrid setups, it might oppose further intertwining among *OpenStack* instances in federated setups. At the same time, replicating all management facilities in each instance introduces additional overhead. Finally, many *CMPs* are proprietary products of public cloud providers that can not be used for self-hosted use cases. However, with Scalr [2] and ManageIQ [1], there are open source based projects that can be customized.

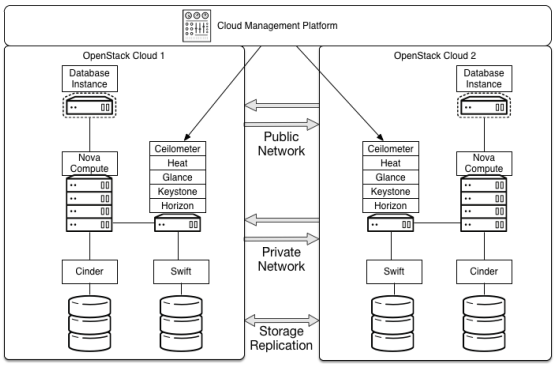


Figure 2: Very few to no alterations to *OpenStack* are required when a *Cloud Management Platform* (CMP) is used to form a federation of multiple instances. Source: [12]

Regarding opportunities for integrating policy language concepts into federated cloud setups, *CMPs* are an oncoming target, since *CMPs* have a similar role compared to *PDPs*. Furthermore, only the *CMP* itself has to be altered. However, the main drawback is that the *CMP* represents a single point of failure. As soon as the *CMP* enters a degraded operational state, the proper enforcement of policies is at stake.

3.1.2 Multi-Site OpenStack Instances

Providing an alternative approach, multi-site *OpenStack* setups consist of multiple *OpenStack* instances, that share a certain set of common services. The *OpenStack* reference design providing location-local services through multi-site installations is illustrated in Figure 3. However, it should be noted that federated setups with different goals can use a very similar setup that only shares the *Keystone* authentication service and does not require a load balancer. When multiple sites are interconnected, *OpenStack* differentiates between four roles that an instance can embody: *Cells*, *Regions*, *Availability Zones* and *Host Aggregates*.

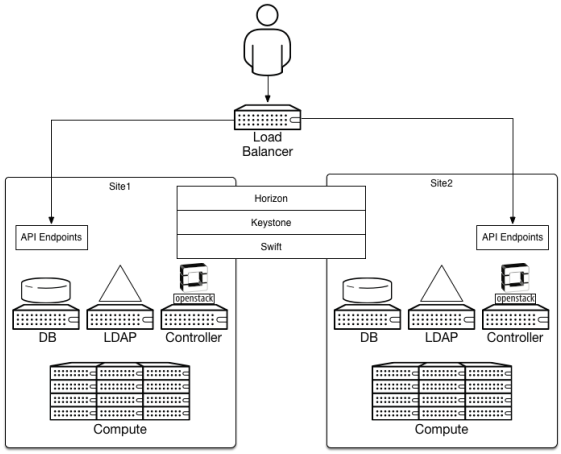


Figure 3: A location-local multi-site setup is illustrated. However, there are many ways to configure a multi-site setup. Many *OpenStack*-components can be shared across sites, however at least the *Keystone* authentication service has to share the same database. Source: [12]

Cells provide the most basic mechanism for organizing a cloud installation in a distributed mode of operation without the need for additional technologies such as *CMPs* or without having to alter existing nova services. *Cells* provide a hierarchical, tree-based structure for partitioning multiple hosts in a cloud setup.
While *Regions* provide similar semantics and are intended for organizing hosts in partitions, the main difference is that *Cells* only expose the API for provisioning compute resources at the top-level *Cell*. In contrast to this, each *Region* exposes its own compute resource API, which provides users with a more explicit mechanism for deciding which region should be used for allocating compute resources.
Both *Cells* and *Regions* provide interesting means for implementing geolocation policy attributes. Using a single entry point at the *Cell* level would be compelling, as the evaluation of policy attributes could be woven into the *OpenStack* nova-scheduler component. However, performing invasive alterations on a quick-moving target such as *OpenStack* comes with a high risk of failure. As a result, using the mechanism of *Regions* seems to be feasible, since the explicit control over regions should make the implementation of a decoupled *PEP* feasible.
Availability Zones can be used to organize *OpenStack* resources in groups that offer a certain degree of physical isolation and/or redundancy from other *Availability Zones*. Providing some examples, the feature can be used to distinguish resources that are connected to a different *Power Distribution Unit* (PDU) on the fine-grained level, or machines that are located in the nearby failover data center. In contrast to this, *Host Aggregates* provide an additional mean for specifying resource domains for load balancing. A popular use case is to use *Host Aggregates* in order to distinguish between different classes of hardware (e.g. processor speed, network link speed, special hardware such as GPUs or FPGAs).
Using the scopes of *Availability Zones* and *Host Aggregates* in multi-site *OpenStack* installations can be leveraged in the implementation of *PEPs* that consider policy attributes such as *QoS* levels or availability parameters.

3.2 Single-node experimental platform for federated OpenStack setups

In this Section, we describe the automated installation process of our OpenStack-based testbed in our virtual environment. We give an introduction to the usage and a conceptual overview. Furthermore, we describe mechanisms for implementing test cases that evaluate the correct behavior of policy enforcement mechanisms using fault injection.
The automated scripts for setting up the testbed are developed and tested using *Ubuntu 14.04.3 LTS (Trusty Tahr)* server version. All required dependencies (e.g., *Ansible*, *libvirt*, etc.) are installed automatically, thus only an Internet connection is required. The complete installation can be started by running a simple script, which creates a federated *OpenStack* setup according to the architecture depicted in Figure 4. On the physical machine (layer Φ), virtual machines are created that represent an *OpenStack Region* or a datacenter location (layer Δ). Within the Δ-layer, further virtual machines are created that host basic *OpenStack* services (layer Ω). It should be noted, that each service runs on a separate virtual machine, which emulates the behavior of a data center, where the services run on individual machines as well. This is also a unique characteristic compared to other single-node *OpenStack* installations such as DevStack [11] or HP Helion [8]. The ι-layer represents virtual machines that are provided to users by *OpenStack*. Finally, applications such as *Hyrise-R* can be run on these end-user VMs on the α-layer. All virtual machines up to the Ω-layer are automatically created in the course of the automated testbed initialization.
The implementation of test cases follows a fixed struc-

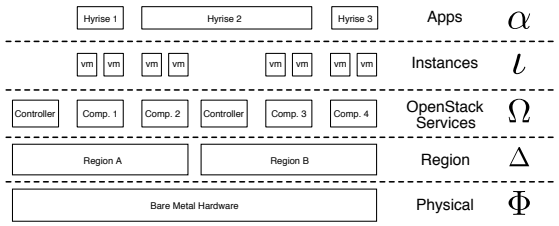


Figure 4: The virtualized *OpenStack*-testbed uses nested virtualization to resemble multiple *Regions* (Δ), individual servers (Ω) within a *Region* that host *OpenStack*-components, and VMs (ι) and applications (α) provisioned in the federated setup.

ture, so that it is easy to add further test cases if necessary. The general idea is that the injection of faults can provide insights on how they affect the functionality of policy enforcement mechanisms and how the implementation of such mechanisms can deal with faults. This serves partly to show weak points of the setup and partly to document details about its behavior. Experiments consist of multiple stages: The *setup* stage creates all elements necessary to run the experiment. The *break* stage then injects a fault. An optional *heal* stage can be implemented to remove the fault. After each of these stages, a *check* step is executed, which observes the state of the system and reports its findings to the user. Generally, after the setup stage all checks should be successful. In cases where automated testing is not feasible and user interaction is required, the user is prompted accordingly.

4 Conclusions

In the upcoming period of the Future SOC lab, we will implement the presented design built around policy language concepts and evaluate its application in the federated cloud scenario. Hyrise-R will be one exemplary application to exploit the presented language to configure policy characteristics, e.g., the replication rate of the stored data. At a lower level of the cloud application stack, we are going to investigate approaches for integrating policy attributes regarding geographical location and *Quality-of-Service* levels into the *OpenStack* ecosystem. To support this process, we are continuing our efforts on building fully virtualized testbed based on OpenStack, which ensures repeatability for both test cases and performance measurements. Furthermore, applying fault injection in test cases will allow us to harden policy enforcement mechanisms even if a degraded system state has been reached. To achieve these goals, we are going to build up on top of our preceeding efforts in the project, where we introduced replication mechanisms to Hyrise [10] and presented a virtualized single-site *OpenStack* testbed that enables us to perform fault injection experiments [4].

References

[1] Manageiq. `http://manageiq.org`. Accessed: 2016-01-14.

[2] Scalr. `http://www.scalr.com`. Accessed: 2016-01-14.

[3] F. Eberhardt, J. Hiller, O. Hohlfeld, S. Klauck, M. Plauth, A. Polze, M. Uflacker, and K. Wehrle. D2.2: Design of inter-cloud security policies, architecture, and annotations for data storage. Technical report, Jan 2016.

[4] J. Eschrig, S. Knebel, and N. Kunzmann. Dependable cloud computing with openstack. In *Proceedings of the Third HPI Cloud Symposium Operating the Cloud 2015*, 2015.

[5] J. Gray, P. Helland, P. O'Neil, and D. Shasha. The dangers of replication and a solution. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, 1996.

[6] M. Grund, P. Cudre-Mauroux, J. Krüger, S. Madden, and H. Plattner. An overview of hyrise - a main memory hybrid storage engine. *IEEE Data Engineering Bulletin*, 2012.

[7] M. Grund, J. Krüger, H. Plattner, A. Zeier, P. Cudre-Mauroux, and S. Madden. Hyrise: A main memory hybrid storage engine. *Proc. VLDB Endow.*, 2010.

[8] Hewlett Packard Enterprise. HPE Helion OpenStack. `http://www8.hp.com/us/en/cloud/hphelion-openstack.html`. Accessed: 2016-01-14.

[9] J. Krüger, C. Kim, M. Grund, N. Satish, D. Schwalb, J. Chhugani, H. Plattner, P. Dubey, and A. Zeier. Fast updates on read-optimized databases using multi-core cpus. *Proc. VLDB Endow.*, 2011.

[10] J. Lindemann, S. Klauck, and D. Schwalb. A Scalable Query Dispatcher for Hyrise-R. 2015. (to appear).

[11] OpenStack Project. Devstack - an openstack community production. `http://docs.openstack.org/developer/devstack/`. Accessed: 2016-01-14.

[12] OpenStack Project. Openstack architecture design guide. `http://docs.openstack.org/arch-design/`. Accessed: 2016-01-12.

[13] H. Plattner. A common database approach for oltp and olap using an in-memory column database. *SIGMOD*, 2009.

[14] D. Schwalb, M. Faust, J. Wust, M. Grund, and H. Plattner. Efficient transaction processing for hyrise in mixed workload environments. In *IMDM in conjunction with VLDB*, 2014.

[15] D. Schwalb, J. Kossmann, M. Faust, S. Klauck, M. Uflacker, and H. Plattner. Hyrise-r: Scale-out and hot-standby through lazy master replication for enterprise applications. In *Proceedings of the 3rd VLDB Workshop on In-Memory Data Mangement and Analytics*, 2015.

One-class Classification for Personal Risk Detection

Jorge Rodríguez, Ari Y. Barrera-Animas, Luis A. Trejo, Miguel Angel Medina-Pérez, Raúl Monroy
Tecnológico de Monterrey, Campus Estado de México
Carretera Lago de Guadalupe Km. 3.5, Atizapán de Zaragoza,
Estado de México, C.P. 52926, México
{jorger, A01373306, ltrejo, migue, raulm}@itesm.mx

Abstract

Personal risk detection aims to discern when a person is experiencing a dangerous situation, looking for deviations from their physiological and behavioural normal patterns. Personal risk detection can be posed as a one-class classification problem, where the classifiers are trained with data obtained by observing the normal activities of the user. In the case of our research, the normal data is obtained using a wearable band, which monitors different health related indicators, such as: hearth rate and skin temperature. Using this data, we constructed a classifier based on a K-means ensemble, which uses random feature projection to add diversity to the classifiers, thus increasing the classification performance. This classifier has an increased accuracy, compared with the ones used in the state of the art, would allow a person in a dangerous situation to more likely and promptly receive aid.

1 Introduction

Currently, there is an increased adoption of consumer-grade wearable devices. Some of these devices provide health monitoring capabilities, which can obtain different user's measures such as heart rate, calorie count or hours of sleep. However, these measurements are not specifically used for each user. The creation of a system that learns from the user's normal patterns of behavior, based on the measurements obtained can lead to detecting when the user is experiencing an anomalous or hazardous situation.

Barrera-Animas *et al.* [2] define Personal Risk Detection as the timely identification of when someone is in a dangerous situation, such as: health crisis, accidents, or other events that may endanger a person's physical integrity. Given that a person may act according to similar behavioral and physiological patterns, with small variations between these, the sensors found in a wearable device may be able to capture a person's normal behavior. Since a risk-prone situation should produce sudden and significant deviations from the person's normal behavior, we can pose personal risk detection as an anomaly detection problem. A one-class classifier can be used to look for anomalies based on a person's normal behavior data.

One of the main goals of our research on personal risk detection was to find a suitable one-class classifier for this problem. The classifier should be able to learn the regularities of the normal behavior from a huge amount of normal behavior records. Since usually a wearable device is paired with a smartphone, it is important that the classifier uses a low amount of memory and computing resources to classify new behavior. This report will discuss the experiments done for this purpose, the results from the experiments and the future work that will be done based on these results.

2 The personal risk detection dataset repository

To have data representative of the scenario where classifiers will work, the *Personal RIsk DEtection* (PRIDE) dataset repository was used. PRIDE contains 23 datasets, each one comprised of the records obtained from observing the health measurements of different users, with diverse characteristics in terms of gender, age height and lifestyle. The test subjects comprised eight female and 15 male volunteers, aged between 21 and 52 years, with heights between 1.56m and 1.86m, and weights between 42 to 101 kg. The volunteers exercising rates ranged from 0 to 10 hours a week, and the time they spent sitting ranged from 20 to 84 hours a week. The health measurements were done using the sensors on the Microsoft Band v1©, recording the values of the sensors using a mobile application developed using the available SDK, and installed in each user's smartphone. The sensors used from the band, and the frequencies of data acquisition for that sensor are described in Table 1. Because the need of charging from the band, each day user's records were not available for about 2 hours. Furthermore, in order to preserve the privacy of the test subjects, they could deattach the device for some activities, such as: taking a shower or participating in water activities.

Table 1: Sensor Descriptions

Sensor	Description	Frequency
Accelerometer	Provides X, Y, and Z acceleration in g units. 1 g = 9.81 meters per second squared (m/s^2).	8 Hz
Gyroscope	Provides X, Y, and Z angular velocity in degrees per second, ($^\circ$/sec) units.	8 Hz
Distance	Provides the total distance in centimeters, current speed in centimeters per second (cm/s), current pace in milliseconds per meter (ms/m).	1 Hz
Heart Rate	Provides the number of beats per minute, also indicates if the heart rate sensor is fully locked onto the wearer's heart rate	1 Hz
Pedometer	Provides the total number of steps the user has taken.	1 Hz
Skin Temperature	Provides the current skin temperature of the user in degrees Celsius.	33 mHz
UV	Provides the current ultraviolet radiation exposure intensity (None, Low, Medium, High, Very High)	16 mHz
Calories	Provides the total number of calories burned by the user.	1 Hz

The data collected from the activities of the user in one week comprises the Normal Conditions Data Set (NCDS), which can be used to construct the normal behavior baseline, which will be used to look for deviations in the behavior, and thus detect risk situations. To test how the users responded to anomalous or risk-prone situations, the same 23 test subjects participated in another data acquisition process, where they needed to perform the following activities: rushing 100 meters as fast as possible, going up and down the stains in a multi-floor building as fast as possible, a two-minute box practice session, falling back and forth, and holding one's breath for as long as possible. Each activity aims to simulate a dangerous or abnormal situation in the real world, e.g., running away from a dangerous situation, evacuating a building during an emergency, fending of an aggressor, swooning, and experiencing breathing problems such as dyspnea. The records of these scenarios comprise the Anomalous Conditions Data Set (ACDS).

2.1 Preprocessing PRIDE for online personal risk detection

Since one of the aims of personal risk detection is the timely detection of risk-prone situations, the raw data from the sensors needs to be converted so that a classifier can recurrently output a decision about the current standing of the user. For that purpose, a feature vector was computed using a window size of one second. Since the frequency of the sensors varied significantly, three cases can be observed for computing the vectors, depending on the readout interval:

Interval less than one second: record the average and sample standard deviation of all the measurements done in a second

Interval equal to one second: record the current sensor value

Interval greater than one second: record the last sensor value

Given these rules, a feature vector contains: the means and standard deviations for the gyroscope and accelerometer measurements; the absolute values obtained by the heart rate, skin temperature, pace, speed, and UV sensors; and the incremental changes (Δ-value) in the absolute values for the total steps, total distance, and calories burnt. A Δ-value was computed as the difference between the current and previous values. Thus, each window of one second results in the 26-dimensional feature vectors with the structure shown in Tables 2 and 3. After preprocessing the data, the NCDS for each user contains in average 320000 records, while the ACDS contains 700 records in average. In order to have statistical validation and also understand how the classifiers performed with different parts of the data, five-fold cross validation was used, by training the classifiers with different 80% of the NCDS and testing with the remaining 20% and the ACDS.

2.2 Testing over PRIDE

The original experiments over PRIDE were reported in Barrera-Animas *et al.* [2]. In that paper, PRIDE contained 18 users, and the performance of the classifiers was measured using:

Precision-Recall (P-R) curves: Precision refers to the fraction of objects correctly classified as belonging to a class, with respect to all the objects classified as belonging to a class. Recall refers to the fraction of objects correctly classified as belonging to a class, with respect to all the objects that belong to a class. In our case, a correctly classified object represents a true anomaly. In addition, Recall is equivalent to the true positive detection rate (TPR). A P-R curve was built for each user independently as well as a single P-R curve based on the mean and standard deviation for all the users.

Receiver Operating Characteristics (ROC) curves: These curves map the TPR versus the false positive detection rate (FPR). Performance indicators were computed based on Fawcett [6]. A ROC curve was built for each user independently and also a single ROC curve based on the mean and standard deviation of the total population. TPR is crucial in a personal risk detection context, where it is preferable to receive several false alerts (false abnormal or dangerous situation) rather than missing one a

Table 2: Feature vector structure (fields 1–18)

Gyroscope Accelerometer						Gyroscope Angular Velocity						Accelerometer					
X axis		Y axis		Z axis		X axis		Y axis		Z axis		X axis		Y axis		Z axis	
$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Table 3: Feature vector structure (fields 19–26)

Heart Rate	Skin						
Temperature	Pace	Speed	UV	$\triangle$ Pedometer	$\triangle$ Distance	$\triangle$ Calories	
19	20	21	22	23	24	25	26

true one (true abnormal situation); hence, it is important to maximize Recall even at the cost of experiencing increasing false alarm rate.

AUC: The AUC of the TPR versus the FPR, which indicates the general performance of the classifier for all FPR rates.

The classifiers that were used to test how viable was to detect anomalous or dangerous situations in PRIDE were:

ocSVM: The implementation of ocSVM [14] included in LibSVM [3] with the default parameter values ($\gamma = 0.038$ and $\nu = 0.5$) and using the radial basis function kernel.

Parzen: Parzen window classifier using the Mahalanobis distance [5]. For every training dataset, the classifier computes the width of the Parzen-window by averaging the distances between objects sampled every 60 s.[1]

k-means1: A version of the Parzen window classifier based on k-means [13]. k-means1 classifies new objects based only on the closest center of the cluster.

k-means2: A version of the Parzen window classifier based on k-means [7]. k-means2 classifies new objects using all the centers of the clusters.

The best results were obtained using ocSVM, with an average AUC for all users of 85% and an standard deviation of 9.8%.

3 Experiments

In order to surpass the results obtained by using ocSVM, we needed to find a classifier that could use the regularities in the data in order to better obtain a normal behavior model, allowing for increased discerning capabilities of anomalies from normal behaviors. If we could not find a classifier that resulted in an increased performance, we would need to create a new one that worked in the context of personal risk detection, taking into account the limitation of the smartphone's memory and computing capabilities. Since the training and testing of different classifiers, over all the users and folds, requires a big amount of computing cycles, we obtained from the Hasso-Plattner-Institut access to a machine with 64 virtual CPUs, 128 gigabytes of memory, and 200 gigabytes of hard disk drive, which we used to test the following classifiers.

3.1 Autoencoders

Autoencoders [8] are an architecture type of neural network, similar to a multilayer perceptron, where the output layer is the same size as the input layer. An autoencoder aims to capture the regularities that make the training data, and using them to transform the data in the hidden layer. After, the output layer tries to reconstruct the input from the values obtained from the hidden layer. An autoencoder can be used as a one-class classifier, obtaining the mean squared error between the output and the input. A low mean squared error means the object to classify is normal, while a high error means it is anomalous. To decide how low or high can the error be without marking the classified object as an anomaly, a threshold is used.

There are variations on autoencoders, that allow for an increased accuracy. The first variation is the denoising autoencoders [16], which corrupt the inputs at training time, and expect the autoencoder to correct the noise at the output . By adding random noise, the denoising autoencoder is more robust to changes in the objects than the normal version. There also exists deep learning autoencoders, called stacked autoencoders [15]. This version uses more hidden layers, each one using as an input, the output of the previous hidden layer, where each layer is trained to detect different regularities in the data, allowing for an increased accuracy of the classifier.

Autoencoders have different parameters that need to be tuned, such as: learning rate, number of neurons in the hidden layer, and epochs to train. Moreover, denoising autoencoder need to adjust the corruption rate and stacked autoencoders the number of hidden layers. Our preliminary experiment consisted of using a genetic algorithm, to obtain the best parameters for

[1] This procedure saved approximately 7 days when computing the distances per test subject using an Intel Core i7-4600M CPU at 2.90 GHz.

training a stacked denoising autoencoder. Each individual of the genetic algorithm encoded a learning rate from .001 to .7, neurons in the hidden layer from 13 to 100, epochs from 50 to 200, corruption rates from 10% to 40% and hidden layers from 1 to 5. For this genetic algorithm, we selected randomly 3 users from the 18 available at the moment, and separated the NCDS into 80% and 20%, which would be used for training and testing. From the 80%, we did again a separation into 80% and 20%, where the former would be used for training in the genetic and the latter for validation. This was done over the five folds. The genetic algorithm fitness function was the mean squared error of all the validation set, where more apt individuals generated a lower mean squared error. The genetic had an 80% probability of crossover and 10% of mutation. All the experiments were implemented in Python, using theano library for the denoising autoencoder, deap library for the genetic algorithm, and scoop library for implementing concurrency to better use all the processors.

The results of our preliminary experiment show an average AUC of 39.7%, with a standard deviation of 7.6. This is at least 45% less than the results of ocSVM. Furthermore, this result is in line with the one obtained using another dataset for anomaly detection using one-class classifiers, but in the domain of intrusion detection. In Rodríguez *et al.* [12], the stacked denoising autoencoders performed worse than ocSVM, but with a difference of 5% instead of 42%. The preliminary experiments indicate that stacked denoising autoencoders may not be suitable for the personal risk detection problem.

3.2 One-Class K-means with Randomly projected features

One of the aims of using stacked denoising autoencoders is that each trained hidden layer detects different regularities in the objects, in order to increase the classification performance. To add diversity, and allow the classifier to learn different regularities in the data, we created the One-Class K-means with Randomly projected features (OCKRA) [11]. OCKRA is an ensemble of 100 classifiers, where each classifier uses a different random projection of the features, according to random subsets of features, which ensure a high diversity among the classifiers [10, 4, 9].

For each classifier, the objects from the projected training set, with a difference between them of 60 seconds are sampled, and the average distance δ_i between all of them is calculated. This distance will serve as the size of each cluster, and will be used to determine the similarity of a new object to a cluster. To obtain the center of the clusters, k-means++ [1] is trained with the projected training dataset, using Euclidean distance (which is standard in previous studies) and $k = 10$. k needs to be small since OCKRA must work online using smartphones, so it should consume low RAM memory and CPU resources while maintaining good classification accuracy.

For classification of an object, for each classifier the object is projected. Then, using Euclidean distance, the distance d is calculated from the projected object to the nearest cluster center. Each classifier outputs a similarity score given by $e^{-0.5(d/\delta_i)^2}$. The final similarity score is computed by obtaining the average of the similarity score given by each individual classifier. Similar to the stacked denoising autoencoder, a threshold is used to determine if the object represents an anomaly or not. If the score is below the threshold, the object is considered normal, while a score equal or above to the threshold means an anomalous behavior.

Using the average of the distance between a sample of objects obtained every 60 seconds, as the maximum distance from the center of a cluster that an object can have, in order to belong to such cluster was tested against two more variants. The first variant is using the distance of the object farthest from the center that belongs to the cluster. The second variant uses the average of the distances of all the objects of a cluster to its center. These three versions of OCKRA were tested using five-fold cross validation and the current 23 users of PRIDE. The best version is the one that uses the average of the distance between all the objects in a sample. This classifier was tested against the classifiers reported in Barrera-Animas *et al.* [2], but using Euclidean distance instead of Mahalanobis, in order to have different comparison points, using the same performance measurements.

4 Results

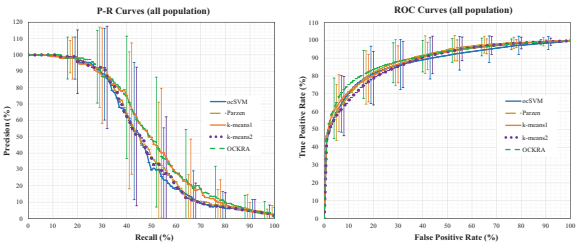


Figure 1: Precision-Recall curves and ROC curves based on the average performance and standard deviation for all users.

Figure 1 shows the average P-R and ROC curves for all the population, where the standard deviations are shown as vertical lines for each algorithm at different intervals. The P-R curves show that OCKRA and k-means1 outperformed Parzen and k-means2, and there was no significant statistical difference between OCKRA and k-means1. The ROC curves confirm the results of the P-R curves, but it can be noticed that OCKRA obtained better FPR rates between 5% and 30%.

Test Subject	ocSVM	Parzen	k-means1	k-means2	OCKRA
Average	86.44	88.56	88.52	86.81	89.09

Table 4: Area (percentage) under the curve for TPR versus FPR.

In order to quantify the differences among the algorithms, the average of AUC results was computed for all the test subjects. Table 4 shows that OCKRA outperformed the other algorithms for at least 0.53% of the AUC on average. This number is small but it has a significant impact because it means that a user will have a higher probability of assistance in a risk-prone situation. Parzen achieved the second best result in terms of AUC but it is less suitable for running on a smartphone because it is two orders of magnitude more expensive than OCKRA (i.e., Parzen requires the full dataset to classify a new object whereas OCKRA requires only 1000 centers of the clusters). In summary, our classifier achieved an AUC above 90% for approximately 57% of the users.

5 Conclusions and Further Work

Personal risk detection is the timely identification of when someone is in a dangerous situation, such as: health crisis, accidents, or other events that may endanger a person's physical integrity. In this research we tested two classifiers for personal risk detection: Stacked Denoising Autoencoders and One-Class K-means with Randomly projected features (OCKRA). The former classifier was an already developed classifier, giving worse results than the state of the art. On the other hand, the latter classifier was developed during this research. Both classifiers were tested using the computing resources obtained from the Hasso-Plattner-Institut. The developed classifier has an increased performance, compared with the state of the art classifiers for personal risk detection. All the experiments were done over the Personal RIsk DEtection dataset repository. OCKRA and the results obtained with this classifier, were published recently in the journal sensors, an can be consulted in [11]
As further work, we want to explore if there is a better number of k centers that will allow us to minimize the memory resources needed to classify, while also maintaining a high accuracy for detecting risk-prone situations. We will also use fast-Fourier transforms, in order to transform our data from the time domain to the frequency domain, in order to obtain more data that can allow us to correctly characterize a person, and thus offer a classifier that has an increased detection of risk-prone situations.

References

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027 – 1035. Society for Industrial and Applied Mathematics, 2007.

[2] A. Y. Barrera-Animas, L. A. Trejo, M. A. Medina-Pérez, R. Monroy, J. B. Camiña, and F. Godínez. Online personal risk detection based on behavioural and physiological patterns. *Information Sciences*, -:–, August 2016.

[3] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1 – 27, 2011.

[4] V. Cheplygina and D. M. J. Tax. *Pruned Random Subspace Method for One-Class Classifiers*, volume 6713, pages 96 – 105. Springer Berlin Heidelberg, 2011.

[5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2001.

[6] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861 – 874, June 2006.

[7] G. Giacinto, R. Perdisci, M. D. Rio, and F. Roli. Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Information Fusion*, 9:69 – 82, January 2008. Special Issue on Applications of Ensemble Methods.

[8] N. Japkowicz. *Concept-learning in the absence of counter-examples: an autoassociation-based approach to classification*. PhD thesis, Rutgers, The State University of New Jersey, 1999.

[9] B. Krawczyk. One-class classifier ensemble pruning and weighting with firefly algorithm. *Neurocomputing*, 150, Part B:490 – 500, 2015.

[10] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2nd edition, 2014.

[11] J. Rodríguez, A. Y. Barrera-Animas, L. A. Trejo, M. A. Medina-Pérez, and R. Monroy. Ensemble of one-class classifiers for personal risk detection based on wearable sensor data. *Sensors*, 16(10):1619, 2016.

[12] J. Rodríguez, L. Cañete, M. A. Medina-Pérez, and R. Monroy. Experimenting with masquerade detection via user task usage. *International Journal on Interactive Design and Manufacturing*, 2016.

[13] D. M. J. Tax and R. P. W. Duin. Combining one-class classifiers. In *Multiple Classifier Systems*, volume 2096, pages 299 – 308. Springer Berlin Heidelberg, July 2001.

[14] V. N. Vapnik. *Statistical Learning Theory*, volume 1. Wiley-Interscience, 1st edition, 1998.

[15] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[16] G. Zhou, K. Sohn, and H. Lee. Online incremental feature learning with denoising autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pages 1453–1461, 2012.

Sequential Anomaly Detection in Business Processes

Christian Linn
AWS-Institute for digitized products and processes
Uni Campus Nord
66123 Saarbrücken
christian.linn@aws-institut.de

Dirk Werth
AWS-Institute for digitized products and processes
Uni Campus Nord
66123 Saarbrücken
dirk.werth@aws-institut.de

Abstract

Many companies use software systems to efficiently manage and control their business processes. As a by-product a huge amount of data is recorded that can be used to understand, model and improve the ongoing processes. One approach is to search for anomalies, i.e. faults and flaws that happen during the process execution. Within this project we investigated data mining techniques to detect anomalies in the sequence of process activities. The techniques are validated on simulated, artificial process data.

1 Introduction

Many companies use information systems, like ERP Systems or Workflow Management System, to control, monitor and manage their business processes. As a byproduct, large sets of transactional data are collected and logged. The availability of such data offers the possibility to obtain insights in existing business processes and to perform data-driven analyses. Of special interest is the detection of anomalies, i.e. flaws and faults in the execution of a business process that can potentially harm the company. While in processes with a predefined and fixed execution path there is only a limited possibility to leave the normal way, the detection of unwanted events is especially important in environments where there is a certain flexibility in how a process can be executed. One approach for an automated analysis of business process data is process mining which adopted data mining techniques to discover, verify, and improve process models [1]. A standard technique to detect business process anomalies with process mining is to discover an initial process model and check its conformance with the process data [2]. A different approach is to use classical data mining techniques to analyze process data for anomaly detection and to avoid the additional step of generating a complex model to describe the underlying process, e.g. in [3]. For an automated analysis of the process data, especially unsupervised data mining techniques are promising as they, in contrast to supervised techniques, do not rely on a training dataset that already contains a classification of the process instances in normal or anomalous.

In the project we investigated the usability of different unsupervised data mining techniques for anomaly detection in the domain of business processes. Four different detection techniques are discussed, that have previously been used for anomaly detection in domains like fraud detection or intrusion detection. The aim of the research is to answer the question whether such methods can in principle be used for anomaly detection in business processes, what is their expected performance and how do they compare in terms of accuracy.

2 Approach

As business processes can be seen as a sequence of activities, the approach of the project was to use sequential data mining techniques for the detection of anomalies. The aim is to give a comparable and quantitative evaluation of basic sequential anomaly detection algorithms in the domain of business processes. According to [4] sequential anomaly detection techniques can be grouped in four different approaches: Kernel based techniques, windows based techniques, Markovian based techniques and methods based on Hidden Markov Models. For this research, we implemented four basic algorithms each representing one of these approaches. All of them operate in an unsupervised way, i.e. without a training sample containing knowledge about the nature (normal or anomaly) of the single sequences. The four techniques were evaluated on an artificial data set simulating a basic business process.

2.1 Kernel based method

The Kernel based method computes a pairwise similarity between all process instances in the data sample. As similarity measure, the normalized length of the longest common subsequence (LCS) between a pair of sequences S_i and S_j is used [5]:

$$nLCS(S_i,S_j) = \frac{LCS(S_i,S_j)}{\sqrt{|S_i||S_j|}}.$$

This similarity measure is commonly used for sequence anomaly detections in other domains, for example in [6]. A k-nearest neighbor (kNN) algorithm is then applied for the point based anomaly detection. As proposed in [7], the anomaly score of each sequence is calculated as the inverse distance to its k-th nearest neighbor.

2.2 Windows based method

For the windows based technique, a sliding window is used to extract subsequences of fixed length l from all process instances. All possible subsequences in the dataset are then, together with their frequency of occurrence, written in a normal dictionary. In a second iteration of the data sample, each subsequences with length l is assigned an anomaly score which is the inverse of the frequency associated with the same subsequence in the normal dictionary. In a last step, the anomaly score of the full sequence A_s is calculated as the sum of the anomaly scores of the subsequences a_s divided by the number of l length windows n_{win} in the sequence [8]:

$$A_s = \frac{\sum a_s}{n_{win}}.$$

2.3 Markovian based method

The Markovian based method estimates the conditional probability of an activity s_i in a sequence $S = (s_1, \dots, s_n)$ based on the previous activities in the sequence. It basically relies on a higher order Markov condition, assuming that the probability for an activity s_i only depends on the previous l activities [9], i.e.:

$$P(s_i|s_1 \dots s_{i-1}) = P(s_i|s_{i-l} \dots s_{i-1}) \text{ for } l > 1.$$

Technically, a sliding window is used to extract all subsequences of length l and l-1 from the data set. Their frequency of occurrence is stored in a normal dictionary. In a second iteration, the conditional probability of each activity in a sequence is calculated as the ratio between the frequency of the subsequences $(s_{i-l}, .., s_i)$ and $(s_{i-l}, \dots, s_{i-1})$:

$$P(s_i|s_1 \dots s_{i-1}) = \frac{f(s_{i-l}, \dots, s_i)}{f(s_{i-l}, \dots, s_{i-1})}.$$

The conditional probabilities of the single activities are then combined to a total probability. To consider the different length of the full sequences, the total probability is normalized to the number of l length windows n_{win} in a sequence to calculate the anomaly score. A higher anomaly score then represents a higher probability for a sequence to be anomalous. A similar approach was proposed in [10].

2.4 Hidden Markov Model

In this method a Hidden Markov Model [11] is constructed which allows to transform the observed activity sequences in the data sample in sequences of n_s hidden states. The Expectation Maximization algorithm is used to perform a maximum likelihood fit to the data sample and determine the parameters of the Hidden Markov Model for the given set of hidden states. After constructing the Hidden Markov Model, the Viterbi algorithm [12] is used to determine the most probable sequence of hidden states for each individual sequence in the data sample. Finally the windows based method as discussed previously is applied to the hidden sequences and a corresponding anomaly score is assigned to each sequence.

3 Used Future SOC Lab Resources

To evaluate and measure the performance of these techniques, a high computational power was required, driven by the complexity of the methods. The computing complexity of some of the investigated detection techniques is proportional to the number of the analyzed sequences, O(n). Some of them however, especially the Kernel Based Method, have a complexity which is up to O(n^2), i.e. goes quadratic with the number of sequences. As we needed many and large datasets to reliably develop and test the algorithms to get accurate results, the main resources required were computing power and storage. Therefore we performed our studies and developments on Future SOC Lab servers were we used resources of up to 400 CPUs and 400 GB storage. The 400 CPUs were distributed in 100 virtual machines. Therefor we also could test and try multithreaded computing with up to 4 CPUs.

4 Findings

The four detection techniques are applied to the simulated business process data sample. As a performance measure to compare the different techniques, we used the ratio d/t, where t is the total number of simulated anomalies in the data sample and d is the number of detected true anomalies in the t instances with highest anomaly score. Table 1 shows the resulting accuracy of the windows and Markovian based methods, tables 2 and 3 the accuracy of the Hidden Markov and Kernel based method, respectively.

Table 1: Accuracy of windows and Markovian based detection techniques for different length of the window and the previous subsequence.

l	Accuracy windows	Accuracy Markovian
1	--	0.96
2	1.0	0.80
3	0.99	0.61
4	0.97	0.30
5	0.93	0.30

Table 2: Accuracy of Hidden Markov Model based method for different number of hidden states in the Markov Model. All numbers are obtained for a windows length of 3 in the subsequent windows based detection method.

n_s	Accuracy HMM
6	0.67
8	0.80
10	0.83
12	0.92
14	0.90

Table 3: Accuracy of the Kernel based method for different choices of the k-th nearest neighbor.

k	Accuracy Kernel
2	0.69
3	0.69
5	0.40

From the results of the simulated sample one can summarize that the windows based method performs best and is most robust with respect to the choice of the parameter value. The Markovian and Hidden Markov Model based methods can reach similar accuracies for an optimized choice of parameters, but show much higher dependency on the parameter settings. The Kernel based method on the other hand does not seem to be a reasonable choice for the anomaly detection in this setting as it is in comparison with the other methods much less performant.

One can conclude that in principle all investigated techniques can be used for sequential anomaly detection in business processes, with the windows based method being the most promising one. The accuracy of the different detection methods also seems to depend on the characteristics of the anomaly that should be addressed. In comparable studies of other domains [7], the windows based method showed a good performance on protein data but compared to other techniques a worse performance for intrusion detection data.

Another important factor for the choice of a suitable detection method is the computing complexity. Especially in real-time conditions this can become a critical performance factor. Amongst the investigated techniques, the Kernel based method is with an O (n^2) complexity by far the most time consuming one.

5 Next Steps

Ideally the methods should be tested on different types of simulated business processes to investigate their performance dependence on the different possible patterns of business process data.

To understand the qualitative difference between the windows based method and the Markov an HMM based methods, a performance study separated for the anomaly types must be performed as a next step in this research.

In addition it is necessary to understand the performance of the discussed sequential detection approaches separately for different types of anomalies and the dependence of the detection techniques on the parameter tuning as well as on the type and complexity of the business process. Especially for more complex processes, future research could investigate the usability of more advanced multi-dimensional detection approaches such as artificial neural networks.

References

[1] W. M. P. van der Aalst, *Process Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[2] W. M. P. Van Der Aalst and A. K. A. De Medeiros, “Process mining and security: Detecting anomalous process executions and checking process conformance,” *Electron. Notes Theor. Comput. Sci.*, vol. 121, no. SPEC. ISS., pp. 3–21, 2005.

[3] M. G. Armentano and A. A. Amandi, “Detection of Sequences with Anomalous Behavior in a Workflow Process,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9262, Q. Chen, A. Hameurlain, F. Toumani, R. Wagner, and H. Decker, Eds. Cham: Springer International Publishing, 2015, pp. 111–118.

[4] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection for discrete sequences: A survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5. pp. 823–839, May-2012.

[5] L. Bergroth, H. Hakonen, and T. Raita, “A survey of longest common subsequence algorithms,” in *Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000*, 2000, pp. 39–48.

[6] S. Budalakoti, A. Srivastava, R. Akella, and E. Turkov, “Anomaly detection in large sets of high-dimensional symbol sequences,” *NASA Ames Res. Center, Tech. Rep. NASA TM-2006-214553*, no. September, 2006.

[7] V. Chandola, V. Mithal, and V. Kumar, “Comparative Evaluation of Anomaly Detection Techniques for Sequence Data,” in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 743–748.

[8] S. A. Hofmeyr, S. Forrest, and A. Somayaji, “Intrusion Detection using Sequences of System Calls,” *J. Comput. Secur.*, vol. 6, no. 3, pp. 151–180, 1998.

[9] D. Ron, Y. Singer, and N. Tishby, “The power of amnesia: Learning probabilistic automata with variable memory length,” *Mach. Learn.*, vol. 25, no. 2–3, pp. 117–149, 1997.

[10] N. Gupta, K. Anand, and A. Sureka, “Pariket: Mining Business Process Logs for Root Cause Analysis of Anomalous Incidents,” *Databases Networked Inf. Syst.*, vol. 8999, no. February, pp. 244–263, 2015.

[11] L. . Rabiner and B. H. Juang, “An introduction to hidden Markov models.,” *IEEE ASSP Mag.*, vol. 3, no. January, pp. 4–16, 1986.

[12] G. D. Forney, “The viterbi algorithm,” *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

Evaluation of a Real-Time Usability Improvement Framework for Business Information Systems

Sharam Dadashnia, Yannick Konrad, Peter Fettke, Peter Loos
Institute for Information Systems (IWi) at the
German Research Center for Artifical Intelligence (DFKI) Campus D3 2, 66123 Saarbrücken
{sharam.dadashnia | yannick.konrad | peter.fettke | peter.loos}@iwi.dfki.de

Abstract

Workflow improvement nowadays plays an important role in the selection process of business information systems. Especially in the context of user-centric development, the usability of such systems is more and more important for the customers. Therefore, in a first step we implement an integrated framework for a dynamic usability improvement for software users and a dashboard for software developers. The paper at hand describes an implementation of the developed concepts and a first step for evaluation using a running example. The implementation is based on an integrated front-end framework called SAP UI5-Framework powered by the In-Memory Database SAP HANA to ensure optimal computation power.

1 Introduction

The project *Real-time Usability Improvement for Business Information Systems* aims at investigating the dynamic workflow improvement of process-based information systems for users (*short term improvement*) and a long term improvement by providing software developers information about the user behavior. This behavior is gathered by logging all single click events from the users. With this information, a software developer gets insights into the behavior, which is otherwise only available by observation of the system usage e. g. by an expert.

The motivation for the investigation is that nowadays the usability of business information systems plays an important role in the selection process of supporting software. This is also true in the context of user-centric development, where the usability of business information systems is a crucial characteristic of differentiation [1]. However, measuring the usability of such systems automatically and their dynamical enhancement and improvement of this systems has not extensively been studied before. The intention is to evaluate an approach, which improves the usability of web-based business information systems in real-time [2]. These developed concepts, which build on data gathering methods from web analytics to provide logging mechanisms for user interactions at a detailed level and subsequently process this data by means of data analytics and process mining methods, are evaluated in an early stage. For the evaluation of the existing concepts, there are certain research objectives and challenges, which have to be processed. An important aspect is to determine which data has to be collected from user interactions. Since a large number of data is logged during the usage of business information systems, an adequate *hardware* and especially *software architecture* is very important. In particular, the following research tasks are investigated:

T1: How can existing concepts be implemented into the SAP UI5-Framework?

T2: Can these concepts generate an (short term) improvement for the users?

Against this background, the remainder of this report is structured as follows: Section 2 describes the chosen research approach. Section 3 elaborates the implementation of the concepts and the integration into the SAP UI5 Framework, before section 4 reports the actually stage of the evaluation of these concepts. Section 4 describes the prototype implementation and section 6 concludes the report and gives an outlook on follow-up projects.

2 Research Approach

The research described in this article is based on the concept of architectural prototyping originating from software architecture development. An architectural prototype in that regard represents a

learning and communication vehicle for the differentiation of styles, features and patterns of a system under development and helps to explore and evaluate the best alternative in the development process [3]. The main objective of the approach relates to problems regarding the application of adequate and specific developed software functions to enable the dynamic improvement in the SAP UI5 Framework. The problem is faced in an iterative manner: by using a repetitive cycle containing a feedback loop, technical approaches and the corresponding programming code are incrementally refined. Figure 1 visualizes the employed four-step research approach with the possibility of multiple iterations in phases two and three, which are executed on the IT basis infrastructure provided by the HPI Future SOC Lab consisting of a SAP HANA In-Memory Database. The software prototype is built on a SAP HANA XS application base.

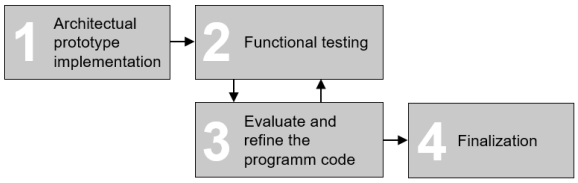


Figure 1: AP research approach

3 Implementation and evaluation of the concepts: preloading of content

To leverage a suitable integration of the developed concepts into the SAP UI5 Framework we extend the existing model-view-controller-pattern from the framework [4]. The model-view-controller-architecture enables the separation of the data models, the presentation and the control flow of applications. The data model includes the necessary application data from the datasets and the exchange between the application and the underlying database. The views of the framework include the graphical representation of the corresponding web pages within the application, which receive the user interaction and provide the requested functionality or data.

To ensure an integrated functionality of the basic UI5 components and the developed code within our framework extension we add a superior controller, a so called *BaseController*. All controllers inherit from the *BaseController* to provide the necessary functions to all underlying controllers and of course the corresponding view components of the software. The *BaseController* in the actual state provides functionality to enable a preloading of content regarding the database connectivity to ensure that provided data from the database is already triggered before the data is actually requested from an accessed view. To visualizes the software architecture figure 2 shows the general overview of the extension of the basic MVC-Pattern of SAP UI5 Framework.

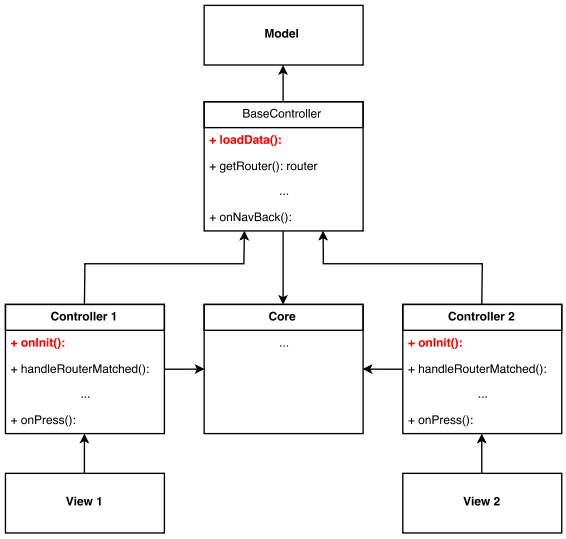


Figure 2: Model-Core-Architecture

As already mentioned, every view component owns a corresponding controller component. In a normal SAP UI5 application, a view calls a corresponding model via the controller. Only data from the corresponding model is available for this view. The model and the controller are bound to the view and they are only visible for this specific view. The main aim of the concept *preloading of content* is to reduce the time which the application take to provide the user with the requested information e.g. a list of available persons for a project staffing decision support. The support is realized by preloading the data about the persons from the database (see figure 3).

The Tool shows a basic functionality of the so called staffing tool. Figure 3 includes two screenshots. On the left side of the screen is a list of persons that are available for staffing a given project.

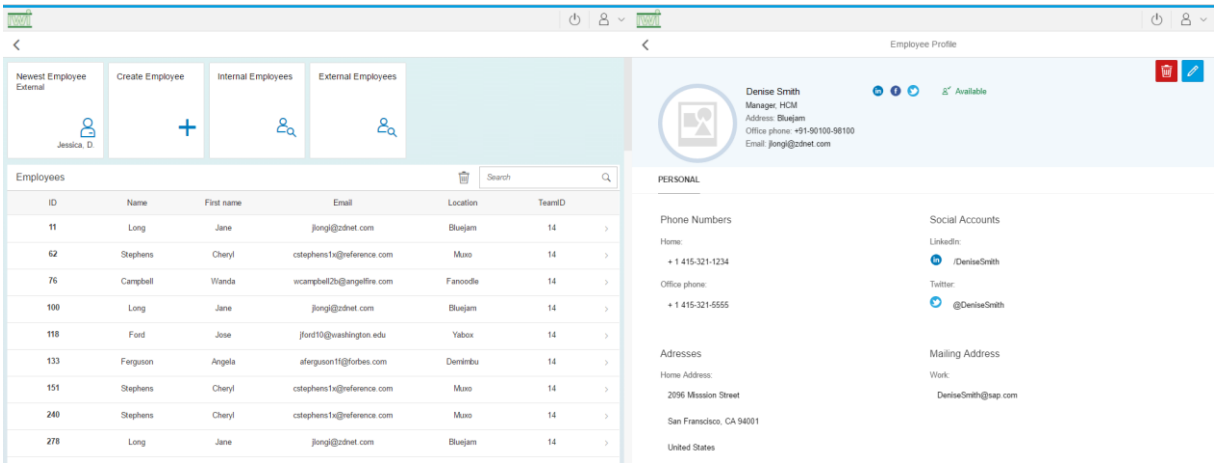


Figure 3: Staffing Tool

The right side shows a detailed profile of a given person in the system. In this running example the list of persons is already preloaded by our software. Running example: View 2 (seen in figure 2) has to load a big amount of data in form of a json-file, which is rendered by the view to ensure a suitable presentation for the system user. View 2 is only accessible by the view 1. The method were using to provide information across certain views is a basic functionality from the sap.ui.core.Core. This function-call boots and prepare the kernel of the SAP UI5-Framework and is available via sap.ui.getCore() in every controller of the application. To load content from the database before the actual view is triggered manually by the user is implemented via the function loadData(). To ensure that the right data is triggered from the database the information about which view and which corresponding controller is called by the method. This information builds the basis for the decision which data has to be preloaded and which not. In the actual state of the prototype, a developer manually configures the code to decide, which data has to be preloaded. The aim is to provide a functionality, which uses information extracted from the user logging to ensure the prediction of the user's next step.

4 Evaluation

We also do a preliminary evaluation of the concept developed within this project period. As mentioned in the running example, the concept *preloading of content* provides the loaded data from the database one step ahead with the main aim to reduce the latency by a screen change for a user. The saved time also reduces the overall process duration and makes processes more efficient. Of course the user has a better "flow" within the application. To give a little overview, we provide a short evaluation of the prototype in the following.

We proved two scenarios once we load 5000 datasets and another scenario we load 35000 datasets from the database. Furthermore, the data

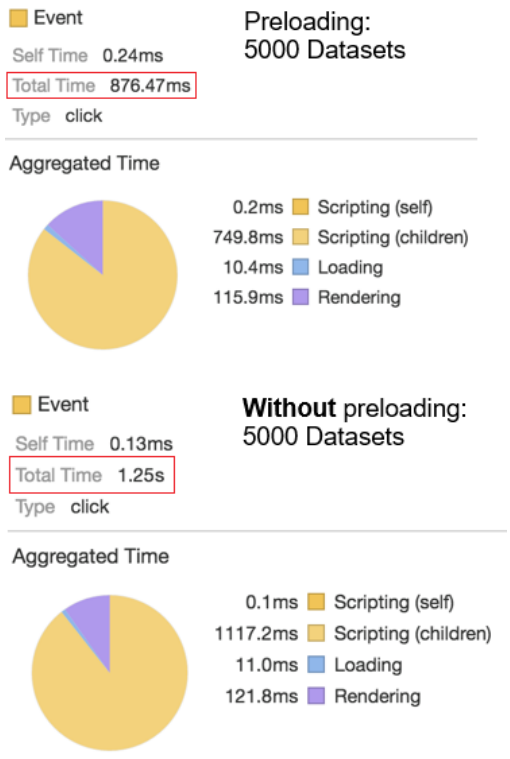


Figure 4: Time reduction with 5000 Datasets

was loaded once without the implemented concept preloading of content and once with the implemented concept. Figure 4 shows the time reduction for 5000 datasets.

The evaluation of the running example with 5000 datasets shows a reduction of 0,37s. Which is about **30%** of the total time. If we increase the dataset, which has to be loaded from the database and rendered on the screen, we could reach the following results seen in figure 5.

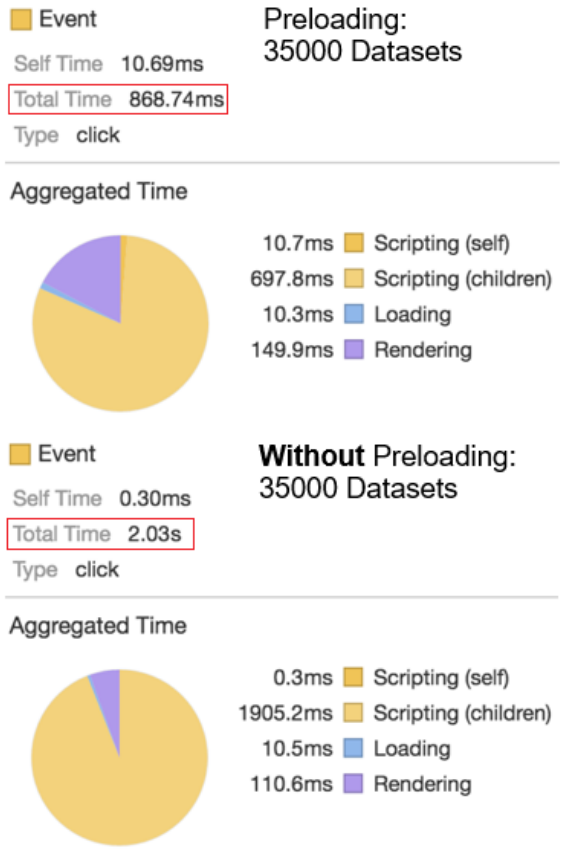


Figure 5: Time reduction with 35000 datasets

The evaluation shows that we can reduce the total time from 2,03s to 0,8687s. There is a total reduction of 1,16s. Which is about **57%** of the total time.

This short evaluation scenario shows that the potential of the implemented method is quite promising. Another aspect is that most of the time – besides the loadData() calls – is spend on rendering the website and parsing the json-model. This is another interesting topic, which is not part of this report at hand but can leverage a lot of time reduction and also provide an improvement of the latency.

5 Provided infrastructure by the HPI

To implement the prototype, we used certain SAP HANA-specific functionalities like stored procedures for data processing as well as the mining of sequences. Since the standard SAP HANA PAL library does not provide the necessary functionality for the calculation, we used the SAP HANA interface to integrate the statistical tool R [5]. The R engine therefore had to be installed on the same server infrastructure provided by the HPI Future SOC Lab to enable calculations based on stored procedures. Furthermore, the R calculation engine was extended by a specialized library called TraMineR [6]. This extension is especially suitable for a first step into sequence clustering and analysis.

6 The BPI Challenge 2016

Furthermore, the given infrastructure is also used to preprocess the log data given by this year's business process intelligence challenge (BPIC'16) [7].

The challenge provides a use case from the Dutch government within the scope of employee insurances and labor market and data services in the Netherlands. The data comprises user interaction data from different IT systems, which is operated by the Dutch Employee Insurance Agency UWV that in turn is commissioned for the implementation and operation of respective services by the Ministry of Social Affairs and Employment (SZW). The challenge was divided into certain questions, which were answered using different techniques like Process Mining, log clustering, a Deep Learning approach and so on [7]. We used the resources from the HPI Future SOC Lab to preprocess the given datasets. Overall, we had datasets containing over 7 million single logs. For joining and exploring the data, the HANA database fits our needs in a perfect way. It enables us to explore the data very fast and give a solid base for the iterations of the analyzing exploration process. One of the results from the challenge were

the so called usage pattern we derived from the given dataset. These usage pattern help to understand the actual behavior within a software application. Using this pattern discovered from the corresponding log files, we derived a pattern seen in figure 6.

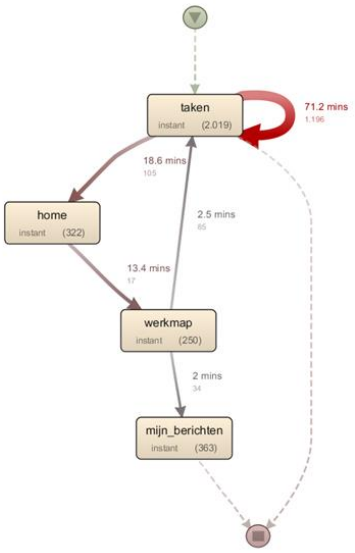


Figure 6: Usage pattern (BPIC'16)

The given usage pattern shows a bottleneck in the section "taken" which corresponds to the tasks for a user. For more information and insights into the data please have a look at the submission [7]. With the report, we submit to the year's BPI-Challenge we reached one of the first places and we were able to present our results at BPM Conference 2016 in Rio de Janeiro.

7 Conclusion and Outlook

The project *Realtime Usability Improvement for Business Information Systems* further implements and extends the fundamental work which has already been conducted in [2]. Within this project period, we implemented parts of the already developed concepts. Besides the extension of the analytical dashboard, we also will develop a tool to support the project management process, a so called staffing tool. This tool constitutes the base for intended evaluations in the futures. The evaluation of the implemented Real-time Usability Improvement Framework described in this article showed that a clear effort in the context of efficiency can be provided by using this techniques. Furthermore, we want to improve the implementation of the running example to create a suitable use case for the comprehensive evaluation of the developed concepts. We also further improved the implemented concepts to leverage more significant dynamic improvements for the software users.

Acknowledgement

The provided high performance IT infrastructure from the HPI allowed the investigation of concrete problem fields in information systems research. The authors thank the HPI Future SOC Lab for the chance of using these resources and appreciate a continuation of the project. The basic concepts were developed in context of the project "Echtzeit Usability Verbesserung auf Basis von Mining-Technologien unter Verwendung von In-Memory Computing" under cooperation with the SAP Innovation Center Potsdam under the supervision of Michael Perscheid, which is funded by the Bundesministerium für Bildung und Forschung BMBF (Software Campus).

References

[1] Lambeck, C., Muller, R., Fohrholz, C., & Leyh, C. (2014, January). (Re-) Evaluating User Interface Aspects in ERP Systems - An Empirical User Study. In System Sciences (HICSS), 2014 47th Hawaii International Conference on (pp. 396-405). IEEE.

[2] Dadashnia S.; Niesen T.; Fettke P.; Loos P.: Towards a Real-time Usability Improvement Framework based on Process Mining and Big Data for Business Information Systems, 2016 In: Tagungsband Multikonferenz Wirtschaftsinformatik. Multikonferenz Wirtschaftsinformatik (MKWI-16), March 9-11, Ilmenau, Germany, 2016.

[3] Bardram, J. E.; Christensen, H. B.; Hansen, K. M.: Architectural prototyping: An approach for grounding architectural design and learning. In: Software Architecture, 2004. WICSA 2004. Proceedings. Fourth Working IEEE/IFIP Conference on (pp. 15-24). IEEE.

[4] SAP SE. Model View Controller (MVC). https://sapui5.hana.ondemand.com/#docs/guide/91f233476f4d1014b6dd926db0e91070.html, viewed 1st October 2016.

[5] The R-Project. Link: https://www.r-project.org/, viewed 3rd October 2016.

[6] TraMineR. http://traminer.unige.ch/, viewed 3rd October 2016.

[7] Dadashnia S.; Niesen T.; Hake P.; Fettke P.; Mehdiyev N.; Evermann J.: Identification of Distinct Usage Patterns and Prediction of Customer Behavior. In: Sixth International Business Process Intelligence Challenge (BPIC'16). Business Process Intelligence Challenge (BPIC-2016), located at BPI / BPM 2016, September 19, Rio de Janeiro, Brazil, Springer, 2016.BPI Challenge

ActOnAir
Sequential Pattern Mining and Classification with SAP HANA

Matthias Scholz
Hochschule Mainz
Lucy-Hillebrand-Straße 2
55128 Mainz
matthias.scholz@hs-mainz.de

Gunther Piller
Hochschule Mainz
Lucy-Hillebrand-Straße 2
55128 Mainz
gunther.piller@hs-mainz.de

Abstract

Goal of the project ActOnAir is the personal guidance of individuals who suffer from asthma and need to reduce their exposure to air pollutants. For this purpose bio-signals and environmental data are captured and analyzed. Applied data mining techniques are sequential pattern mining and classification with decision trees. This contribution describes details of the implementation of the mining components with SAP HANA.

1 Introduction

The project ActOnAir has already been introduced in previous reports. A first paper focused on the research question and the architecture of the overall IT system [1]. The data mining methods were outlined in a second report [2]. A brief summary of these topics is added in this publication for completeness.

Focus of ActOnAir is the personal guidance of asthma patients to reduce their risk of asthma attacks. For this purpose health factors and the environmental exposure of individuals are captured in a comprehensive and fine granular way. Based on the analysis of these data personal guidance shall be provided.

The corresponding ActOnAir IT system consists of the following major components [1, 2]: A mobile sensor box for capturing the individual exposure of persons to air pollutants, a sensor integration and geo sensor network for the processing and storage of heterogeneous sensor data, a data mining and forecasting module for the derivation of a forecast model and a mobile application for the recording of personal health symptoms and the provisioning of real-time forecasts about health risks.

The method and architecture for data mining were outlined in [2]. In brief: ActOnAir follows the pattern based decision tree method of Lee et al. [4]. Here sequential pattern mining is used for feature selection. Features are then used for classification with decision trees.

In this report implementation details of the data mining components are described.

2 Pattern Identification

First, the sequential order of environmental data and health symptoms before days with asthma attacks and days without discomfort need to be captured.

Input

Dataset D_t, containing all environmental factors and individual bio-signals in temporal order with discretized values for one person or for a characteristic patient segment.

Columns in D_t: DATE, PROPERTY, PERSON, VALUE

Here PROPERTY specifies the measurement or sensor type, e.g. temperature, ozone or particulate matter.

Output

Needed are sequences $S = \langle s_1, s_2, \ldots s_n \rangle$ of temporally ordered values for environmental factors and individual bio-signals with a specified length n before days with and without asthma attacks. For example, a sequence of length three for temperature contains the discretized temperature values for three subsequent days, e.g. $\langle cold, medium, high \rangle$.

All sequences with length n before days with an asthma attack are collected in a dataset for high-risk sequences D_h. Sequences of similar length before days without discomfort build a dataset D_l of low-risk sequences.

Columns for D_h, D_l: SEQUENCE_ID, DATE, DAY_ BEFORE, PROPERTY, PERSON, VALUE

Functional outline

To collect all sequences for the dataset of high-risk sequences D_h, the following steps are required:

- Identify all days with discomfort and the corresponding n days before (e.g. $n = 3$ *days = length of sequence)* for all persons within a segment.
- For the identified days store the date and value for all measurements, i.e. properties.
- Persist the results in D_h.

Implementation

All measured sequences for different properties are stored in D_t, which is used as starting point. To capture all days with an asthma attack, one selects from D_t all days where the individual bio-signal ASTHMA ATTACK equals TRUE into a virtual table D_e and assigns a unique ID.

Now one can select the n days before the event and obtain the values for all properties and persons for these days with just one single statement.

To create the offsets to obtain all relevant days before an attack, the ADD_DAYS function is used in a sub select statement (Figure 1, line 10). The result is cross joined with the dates of the days with discomfort in D_e (Figure 1, line 14-19).

```
1  result_High_Risk =
2  SELECT day_before AS "DATE",
3      dates."EVENT_ID" as "SEQUENCE_ID",
4      dates.off as "DAY_BEFORE",
5      timeseries."VALUE",
6      timeseries."PROPERTY",
7      timeseries."PERSON"
8  FROM
9  (
10    SELECT event."DATE", dateRange.OFFSET,
11    ADD_DAYS(event."DATE", dateRange.OFFSET) AS
12    day_before
13    FROM "DATASET_EVENT" event
14 CROSS JOIN
15 (
16    SELECT 0 AS OFFSET FROM DUMMY UNION ALL,
17    SELECT -1 AS OFFSET FROM DUMMY UNION ALL,
18    SELECT -2 AS OFFSET FROM DUMMY
19 ) dateRange
20 ) dates
21 INNER JOIN
22 (
23    SELECT "DATE", "VALUE", "PROPERTY", "PERSON"
24    FROM "DATASET_TIMESERIES"
25 ) timeseries
26 ON dates.day_before = timeseries."DATE"
27 AND dates. "PERSON" = timeseries."PERSON";
```

Figure 1: Pattern Identification

In this way one has obtained a list of dates with discomfort and, for $n = 3$, the three days before. With an inner join one can then select all measured properties and their corresponding values in D_t (Figure 1, line 21-27). This join is not just based on the DATE column but also on the PERSON column (Figure 1, line 26-27). Thus it captures all sequences for all persons of one segment in one step.

As result one receives a list of all sequences with a length of three days for an asthma attack for all properties.

To get the dataset D_l of low-risk sequences before days without attack, the same approach is used. Before all records which are used in D_h are eliminated from the dataset D_t, since these days are not part of a low-risk sequence. To find all days without asthma attack one could select from D_t all the days where the individual bio-signal ASTHMA ATTACK equals FALSE into a virtual table D_e. But, since after the elimination described above, all remaining days in D_t are days without asthma attack, the reduced D_t now equals D_e.

After that one can execute a similar procedure as for the high-risk case to obtain all needed sequences before days without an asthma attack.

Performance

The implementation leads to a high level of parallelization. All sequences for all properties and all persons belonging to one segment are identified in one step – independent from the number of persons and properties.

Initial performance tests were carried out with generated datasets. Each dataset contains 1 to 1000 persons. For each person 365 records for each property were generated. Each record stands for one measured value of a certain day. A setup with 1 or 5 properties was chosen.

As expected, the results in Table 1 show that there is no correlation between the runtime and the number of persons or properties.

Properties	1		5	
Persons	**Records**	**Runtime**	**Records**	**Runtime**
1	730	< 00:01	2.190	< 00:01
5	3.650	< 00:01	10.950	< 00:01
10	7.300	< 00:01	21.900	00:01
25	18.250	< 00:01	54.750	00:01
50	36.500	00:01	109.500	00:02
100	73.000	00:01	219.000	00:03
250	182.500	00:02	547.500	00:05
500	365.000	00:03	1.095.000	00:08
1000	730.000	00:04	2.190.000	00:10

Table 1: Pattern Identification

3 Sequential Pattern Mining

Starting out from the identified patterns before days with and without asthma attacks, patterns with high frequency have to be found.

Input

Datasets D_h and D_l

Columns in D_h, D_l: DATE, SEQUENCE_ID, DAY_BEFORE, PROPERTY, PERSON, VALUE

Output

Datasets D'_h and D'_l

Columns in D'_h, D'_l: SEQUENCE, PROPERTY, SUPPORT, CONFIDENCE, LIFT

Functional outline

To find sequences with high frequency, sequential pattern mining is carried out. Mining has to be performed separately for all measurement types, i.e. properties, within the sets of high- and low-risk patterns, based on the corresponding datasets D_h and D_l, respectively.

Currently SAP HANA PAL does not provide an algorithm for sequential pattern mining. Therefore an R implementation of the cSPADE algorithm has been used [5].

Implementation

The R code is embedded in form of a corresponding RLANG procedure in SAP HANA. Its execution is carried out in an external R environment. The input sequences are passed to the RLANG procedures through virtual tables. These are then transformed into appropriate R data frames. Those data frames are then read into corresponding transaction objects [6]. They serve as final input for the cSPADE algorithm [5].

Performance

Currently no parallelization is used here, since for the moment performance of pattern mining is not a major issue. If the current approach would be extended to include self-learning methods for feature identification, also parallelization of this step should be considered [7].

Performance tests for this component are based on the datasets outlined in Section 2. As expected Table 2 shows that the runtime increases linearly with the number of properties and persons.

Properties	1		5	
Persons	**Records**	**Runtime**	**Records**	**Runtime**
1	730	< 00:01	2.190	< 00:01
5	3.650	< 00:01	10.950	00:01
10	7.300	< 00:01	21.900	00:02
25	18.250	00:01	54.750	00:06
50	36.500	00:03	109.500	00:12
100	73.000	00:05	219.000	00:23
250	182.500	00:12	547.500	01:00
500	365.000	00:26	1.095.000	02:12
1000	730.000	00:50	2.190.000	04:14

Table 2: Sequential Pattern Mining

4 Decision Tree Mining

Based on the sequential patterns, decision trees are built. They allow a classification of actual observations into situations with high- or low-risk for asthma attacks.

Input

Datasets D_h, D_l and D'_h, D'_l

Columns in D_h, D_l: DATE, SEQUENCE_ID, DAY_BEFORE, PROPERTY, PERSON, VALUE

Columns in D'_h, D'_l: SEQUENCE, PROPERTY, SUPPORT, CONFIDENCE, LIFT

Output

PMML models for decision trees

Functional outline

The identified frequent sequential patterns for different properties are interpreted as features. On the other hand, the captured sequences of measurements before days with and without asthma attacks are considered as transactions. These can now be characterized by the presence or absence of identified features – which may be interpreted as attributes of the transactions. Note, that one attribute describes whether an input sequence belongs to a high- or low-risk dataset, respectively. For illustration, a table for the feature characterization is shown in Figure 2.

The following steps are carried out:

- Creation of a table which contains all features as columns.
- Separation of transactions from high- and low-risk datasets.
- Characterization of transactions by the presence or absence of identified features.
- Storage of results in a dedicated table.

SID	Sequence	Temp. Pattern 1 [m, m]	...	Temp. Pattern n [m,c]	Attribute x Pattern n	...	Attribute y Pattern n	Is High Risk?
1	[m, m, c]	1	...	1	0	...	0	Yes
2	[c, m, m]	1	...	0	0	...	0	Yes
...	...	...	...	...	...	...	...	...
n	[m, c, h]	0	...	1	0	...	0	No

Figure 2: Transactions and Features

Implementation

For the creation of the table which contains one column for each frequent sequential pattern, dynamic SQL is used. This allows the construction of appropriate SQL statements during the execution of procedures. When iterating over all frequent patterns, the corresponding properties are added as consecutive numbers used as column names for SQL constructs. This is done separately for high- and low-risk frequent patterns, respectively.

For high-risk sequences all transactions are selected from D_h. SEQUENCE is then inserted as TRANSACTION, while PROPERTY and the value TRUE for "high-risk" are denoted in the attribute table. Sequences in D_l are processed in a similar way, by replacing the value for the risk segment TRUE by FALSE.

Dynamic SQL is also used for the following steps. An update statement is created, where the feature column and the conditions, characterizing the presence or absence of features in transactions, are created dynamically. For the conditions CASE WHEN is used. With LOCATE transactions and features are compared, storing 0 for the absence and 1 for the presence of a feature within a transaction. The corresponding code extract is shown in Figure 3.

```
1  UPDATE att_tbl
2  SET att_tbl."Attribut_1_1" =
3  trans_tbl."Attribut_1_1",
4      att_tbl."Attribut_1_2" =
5  trans_tbl."Attribut_1_2",
6      att_tbl."Attribut_2_1" =
7  trans_tbl."Attribut_2_1",
8  FROM (SELECT
9    CASE
10     WHEN (LOCATE("TRANSACTION", 'FEATURE_1') > 0
11     THEN 1
12     ELSE 0
13   END AS "Attribut_1_1",
14   CASE
15     WHEN (LOCATE("TRANSACTION", 'FEATURE_2') > 0
16     THEN 1
17     ELSE 0
18   END AS "Attribut_2_1",
19   CASE
20     WHEN (LOCATE("TRANSACTION", 'FEATURE_3') > 0
21     THEN 1
22     ELSE 0
23   END AS "Partikel_136",
24   "TRANSACTION",
25   "PROPERTY"
26 FROM "TRANSACTION_TBL") trans_tbl,
27 "ATTRIBUT_TBL" att_tbl;
```

Figure 3: Decision Tree Mining

The transactions and their corresponding attributes are then taken as input for decision tree mining. For this purpose the CART algorithm from PAL [8] is used. Target for the mining process is the attribute describing whether a dataset belongs to the high- or low-risk segment. The resulting tree itself is a binary tree. Each node queries the presence of a frequent sequential pattern.

Performance

The performance of this part can be illustrated by the dataset from Section 1. Examples for results are shown in Table 3.

The results show a growing runtime when increasing the number of records. The reason is that with a rising number of records the number of sequences increases proportionally and more rows need to be characterized through features.

Properties	1		5	
Persons	**Records**	**Runtime**	**Records**	**Runtime**
1	730	00:02	2.190	< 00:01
5	3.650	00:02	10.950	00:01
10	7.300	00:03	21.900	00:02
25	18.250	00:03	54.750	00:06
50	36.500	00:03	109.500	00:12
100	73.000	00:05	219.000	00:23
250	182.500	00:08	547.500	01:00
500	365.000	00:10	1.095.000	02:12
1000	730.000	00:18	2.190.000	04:14

Table 3: Decision Tree Mining

5 Next Steps

Since August 2016 the ActOnAir iOS app is available. This will enable tests of whole IT system with realistic data – including the mining and forecasting components.

In addition to that a runtime for forecasting, based on the derived classification models, will be implemented upon SAP HANA. Systematic model improvements can then be studied by using different sets of training and test data. Here in particular various pruning strategies for feature selection shall be investigated.

Finally, it will be interesting to compare sequential pattern mining on R with pattern mining through a new PAL component, which is planned for HANA 2.0, SPS0.

Supported by the Federal Ministry for Economic Affairs and Energy

References

[1] Scholz M., Bock N., Piller G., Böhm K.: ActOnAir: Data Mining and Forecasting for the Personal Guidance of Asthma Patients. In: Proceedings HPI Future SOC Lab Day Fall 2015. HPI Future SOC Lab, Potsdam, Germany, Universitätsverlag Potsdam, 2015

[2] Scholz M., Piller G.: ActOnAir: Data Mining of Environmental Data and Bio-signals. In: Proceedings HPI Future SOC Lab Day Spring 2016. HPI Future SOC Lab, Potsdam, Germany, Universitätsverlag Potsdam, 2016

[3] Bock N., Scholz M., Piller G., Böhm K., Müller H., Fenchel D., Sehlinger T., van Wickeren M., Wiegers W.: Systemarchitektur eines mobilen Empfehlungssystems mit Echtzeitanalysen von Sensordaten für Asthmatiker. In: Proceedings MKWI 2016 Research-in-Progress: 23-29, 2016

[4] Lee C. H. et al.: A Novel Data Mining Mechanism Considering Bio-Signal and Environmental Data with Applications on Asthma Monitoring. Computer Methods and Programs in Biomedicine, 101 (1): 44-61, 2011

[5] Buchta C., Hahsler M., Diaz D.: Package 'arulesSequences'. https://cran.r-project.org/web/packages/arulesSequences/arulesSequences.pdf, 2016. Last accessed 30th September 2016

[6] Hahsler M. et al.: Package 'arules'. https://cran.r-project.org/web/packages/arules/arules.pdf, 2016. Last accessed 17th September 2016

[7] Mofor G.: Parallelization options with the SAP HANA and R-Integration. http://scn.sap.com/docs/DOC-71696, Last accessed 22th September 2016

[8] SAP PAL: SAP HANA Predictive Analysis Library. http://help.sap.com/hana/sap_hana_predictive_analysis_library_pal_en.pdf, 2016. Last accessed 11th March 2016

Optimizing the Utilization in Cellular Networks using Telenor Mobility Data and HPI Future SoC Lab Hardware Resources

Julia Sidorova, Lars Lundberg
Blekinge Institute of Technology
Karlskrona, Sweden
julia.a.sidorova@gmail.com
lars.lundberg@bth.se

Lars Sköld
Telenor Sweden AB
Stockholm, Sweden
Lars.Skold@telenor.se

Abstract

The revenue of a cellular network is proportional to the number of subscribers that can use it without overloading any radio cell. The mobility pattern of the subscribers affects the load in the network. Based on data from a region in Sweden, we have evaluated two optimization strategies: Tetris optimization and cell expansion. Tetris optimization tries to find the mix of users from different market segments that provides the most even load in the network. Cell expansion is done by selectively expanding the capacity of heavily loaded radio cells using cell splitting. Both optimization strategies are based on linear programming (LP), and HPI Future SoC Lab hardware resources have been used for solving these LP problems.

1 Project idea

We want to increase the utilization in a cellular radio network. The potential revenue of the radio network is proportional to the number of subscribers that can use it without suffering from quality problems due to overloaded radio cells. The mobility pattern of the subscribers, i.e., where they tend to be during different times of the week, affects the load in the radio cell network. A dream scenario is to have an even geographical spread of subscribers during all hours of the week, because then all cells are equally loaded during all the time and we get a very high utilization of the physical infrastructure. This kind of optimal spread of the load is in most cases not possible, but still a goal for the operator. In order to maximize the value of the cellular radio network, the initial planning of the network tries to predict the mobility pattern of the subscribers by having a large number of small cells in city centers, sports arenas, and other places where one can expect a high density of subscribers during some time periods.

The marketing department of a telecom operator often divides the market into user segments, e.g., young adults, families, and business men/women. One of the reasons of dividing the market into such segments is that different marketing campaigns can target these groups. If the subscribers in different segments have different mobility patterns, and if these patterns complement each other in the sense that subscribers from one segment tend to be at different locations than subscribers from some other segment, then these two segments would be a good and complementary mix from an infrastructure utilization point of view. If we know the (average) mobility pattern for subscribers in such segments, this information could be used for finding a mix of subscriber segments that maximizes the utilization of the radio network. We will use the term *Tetris optimization* (the name is inspired by the famous game where one should combine complementary shapes) for the process of finding a mix of user subscriber segments that maximizes the utilization of the radio network (we will define this process in detail later).

Tetris optimization is one way to increase the number of subscribers in a radio cell network without risking quality problems due to overloaded cells. Provided that the mobility pattern of the different subscriber segments are known this approach can be explored by the marketing department by targeting different market segments. If the mobility patterns of the subscribers are known, we could also use another approach for optimizing the infrastructure utilization. That approach is to do selective expansion of the radio network based on observed hotspots, i.e., one can insert new radio equipment and split a heavily loaded cell into smaller cells, thus making it possible to increase the number of subscribers without risking quality problems due to overloaded cells; we call this approach *cell expansion*. Cell expansion is done by the technical department and does not require any selective marketing activities from the marketing department.

One could of course also combine the Tetris optimization and cell expansion approaches. One way of doing this is to first apply Tetris optimization and then identify and expand the radio cells that tend to

be the bottlenecks for the optimized subscriber mix. Another way to combine the two approaches is to expand the radio cells that are the bottlenecks in the default mix of subscribers, and then do Tetris optimization on the expanded network.

A database provided by Telenor, and used in this study, contains historical location data from a region in Sweden with more than 1000 radio cells during one week in 2015 with the user's location registered every 5 minutes. This means that we have 7*24*12 = 2016 time slots of 5 minutes each. There are 27010 unique subscribers in the database.

The marketing department has identified six user segments:

1. Corporate clients (139 subscribers)
2. Cost aware, (4003 subscribers)
3. Modern John/Mary, (5963 subscribers)
4. Quality aware, (5805 subscribers)
5. Traditional, (6007 subscribers)
6. Value aware, (5093 subscribers)

Tetris optimization and cell expansions are both based on linear programming (see sections 1.2 and 1.3).

1.1 Related work

Analyzing mobile traffic has become increasingly important. In [17] Naboulsi et al provide a survey of studies using data collected by mobile operators. One of the findings in that survey is that users, and user segments, tend to follow patterns and visit the same locations during the same periods of the week.

There are two main areas of related work that are relevant for this study: base station placement and other forms of infrastructure optimization similar to cell expansion (see Section 1.1.1), and geodemographic user segments such as the ones used by Tetris optimization (see Section 1.1.2).

1.1.1 Optimization of the Physical Infrastructure

Optimizing wireless radio cell networks has been studied for a long time [19][12][20][21][16][15][11]. In [3] the authors investigate different mathematical programming models for deciding where to install new base stations and how to select their configuration, to find a trade-off between coverage and cost; similar problems have been addressed in [26] and [2]. The concept of force fields, motivated by the physics of multiple particles in a closed system, has also been used for optimizing base station placement [18].

Optimizing cell planning in modern radio networks with mixed cell sized is a challenging problem. In [22] and [4], the authors investigate how genetic and other optimization algorithms can be used for finding good locations for base-stations in networks with mixed cell sizes. Optimized planning of heterogeneous radio networks, where small cells are deployed within large macrocells has been studied by Wang et al [23]. The challenge in this case is to find a cost-effective way to satisfy the traffic requirements of the users.

The optimal placement of base stations and relay stations in WiMAX (IEEE 802.16) networks has been studied by Yu et al [27]. In that paper the authors define a model that uses integer programming to find optimal physical locations of base stations and relay stations in IEEE 802.16j networks. In [1] the authors extends the study by Yu et al by allowing relay stations to be located several hops away from the base station. An algorithm for optimal relay and base station placement has also been developed by Islam et al [13]; González-Brevis et al have looked at base station placement for minimal energy consumption [9].

1.1.2 Geodemographic User Segments

Geodemographic classification is now used by almost all large consumer-oriented commercial organizations to improve their understanding of the appeal of their products and services to different market segments. The two major segmentation systems are ACORN (a classification of residential neighbors) developed at CACI Limited and MOSAIC developed by CNN marketing. However, there are commercially available systems by other companies, detailed down to the postcode level. One of the reasons segmentation systems like ACORN are so effective is that they are created by combining statistical averages for both census data and consumer spending data in predefined geographical units [8]. Originally developed for the UK, MOSAIC used some 400 items of small area information to classify each of the 1.3 million UK postcodes into 61 residential neighborhood types. The postcode descriptors allow us powerful means to unravel lifestyle differences in ways that are difficult to distinguish using conventional survey research given limited sources and sample size constraints [25]. It was demonstrated that middle-class categories in the UK such as 'New Urban Colonists', 'Bungalow Retirement', 'Gentrified Villages' and 'Conservative Values', whilst very similar in terms of overall social status, nonetheless register widely different public attitudes and voting intentions, show support for different kinds of charities and preferences for different media as well as different forms of consumption. Mosaic categories also correlate to diabetes propensity [14], school students' performance [24], broadband access and availability [8] and so on. Industries rely increasingly on geodemographic segmentation to classify their markets when acquiring new customers [10]. The localized versions of MOSAIC have been developed for a number of countries, including the USA, Australia, Sweden, Spain, Germany, and Norway. The main geodemographic systems are in com-

petition with each other (e.g. Claritas, CACI, MOSAIC, etc), and the exact details of the data and methods for generating lifestyles segments are never released [6]. As a result, the specific variables or the derivations of these variables are unknown.

1.2 Formulation of the Linear Programming (LP) Problem

We would like to maximize the number of subscribers under the restriction that the number of subscribers in any cell during any five minute interval does not exceed the cell capacity. The total number of subscribers in segment i is denoted S_i. The number of subscribers belonging to segment i in cell j during time slot t seen from the database is denoted $a_{i,t,j}$. The maximum subscriber capacity in cell j is denoted C_j. For each subscriber segment we introduce a scaling coefficient x_i; we optimize these scaling coefficients in our linear programming problem. The existing mix of subscribers corresponds to $x_i = 1$ ($1 \leq i \leq 6$). If we change some x_i, we assume that the number of subscribers in each cell at each point in time will change proportionally.

Given this notation and assumptions, the linear programming problem becomes:

Maximize $\sum_{i=1}^{6} S_i x_i$

Subject to $\sum_{i=1}^{6} a_{i,j,k} x_i \leq C_j \; \forall j$ and $x_i \geq 0 \; \forall i$

During Tetris optimization the capacity C_j is the same for all cells (we will use different C_j for different cells when we combine Tetris optimization with cell expansions). The capacity is selected as the maximum number of subscribers seen in any cell during any 5 minute time slot. For our data set $C_j = 165$. The relative gain of Tetris optimization is not affected by the absolute value of C_j, i.e., we would get the same relative gain for a larger C_j.

In some cases one may only be interested in ways to expand the current number of subscribers, i.e., one would not like to get rid of any subscribers irrespectively of their segments. In the Tetris optimization this corresponds to all scaling factors being larger than or equal to one, i.e., we need to add the restriction that $x_i \geq 1 \; \forall i$. In this case the cell capacity C_j affects the gain of doing Tetris optimization. For $C_j = 165$ we get no gain in this case, but for larger C_j we will see a gain.

1.3 Cell Expansions

The capacity of a radio cell can be expanded by inserting new hardware and using conventional methods such as cell splitting, e.g., splitting the old cell into two or more new cells. In fact, cell splitting is an important technology in order to achieve network densification, which is a key mechanism for 5G networks [5]. If we split an old cell into two new and are able to do a perfect split, half of the users in the old cell will end up in each of the two new cells. The split would probably be able to cut the geographical area cover by the old cell into two (almost) equally sized cells. During the peak hours there are probably active phones in almost all parts of the cell, i.e., one could argue that splitting the load during the peak periods into half is optimistic, but not completely unrealistic. If we make the (extremely) pessimistic assumption that the load in a certain part of a cell is totally unrelated to the size of that part, we will on average have 3/4 of the original load in the most heavily loaded cell after the split. Unless explicitly stated otherwise. In order to strike a compromise between the optimistic and the pessimistic assumptions, we will assume that the subscribers in each of the two new cells is at most 2/3 of the number of subscribers in the old cell; the 2/3 assumption corresponds to multiplying the capacity of the original cell with a factor 3/2. The optimistic assumption that the load is split into two equal halves corresponds to multiplying the capacity in the original cell with a factor 2; the pessimistic 3/4 assumption corresponds to multiplying the capacity in the original cell with 4/3. Obviously, expanding a cell affects the capacity in all time slots. This means that expanding cell number k corresponds to multiplying the cell capacity C_k with a factor 3/2 in our linear programming model. When doing pure cell expansions we do not want to do Tetris optimization, i.e., we want to increase the number of subscribers, but not change the mix of subscribers. In order to keep the mix of subscriber segments we add the restriction $x_1 = x_2 = x_3 = x_4 = x_5 = x_6$ to our linear programming model.

2 Used Future SOC Lab Resources

We continued to use the same hardware as previously in the HPI Future SOC Lab. The hardware was used to solve the linear programming problems discussed in the previous section. We used the Gurobi solver [7] for the LP problems. Solving these LP problems is very computationally demanding and the computation time for generating graphs like the ones in figures 1 and 2, is around 4-12 hours using modern hardware. Having access to the powerful hardware resources at HPI Future SoC Lab has therefor been very important for us.

3 Findings

3.1 Tetris Optimization

As mentioned in Section 1, there are 27010 subscribers in the data, and the cell capacity is set to 165 for all cells, which is the minimum cell capacity for handling the data set. When solving the optimization problem, we get an objective function value of 42755 subscribers. This corresponds to a 58% increase of

the number of subscribers using the same physical radio network (42755/27010 = 1.58). The x-values (scaling factors) that yields the optimal result are: $x_1 = 0$, $x_2 = 0.13$, $x_3 = 0$, $x_4 = 1.45$, $x_5 = 4.85$, $x_6 = 0.92$.

3.2 Cell Expansions

Fig. 1 shows the number of subscribers as a function of the number of cell expansions. We have cell expansion added the restriction that $x_1 = x_2 = x_3 = x_4 = x_5 = x_6$ in our linear programming model. The active restrictions can be easily identified and each restriction is related to a radio cell. In some cases there are more than one cell preventing us from adding more subscribers. This can be seen as flat segments in Fig. 1. The figure shows that 100 cell expansions increases the maximum number of users from 27,000 to more than 100,000 when we use the expansion factor 3/2. For the expansion factors 2 and 4/3 we see that the difference in terms of the maximum number subscribers increases when the number of cell expansions increases. The reason for this is that for the lower expansion factors, more cells need to be expanded multiple times.

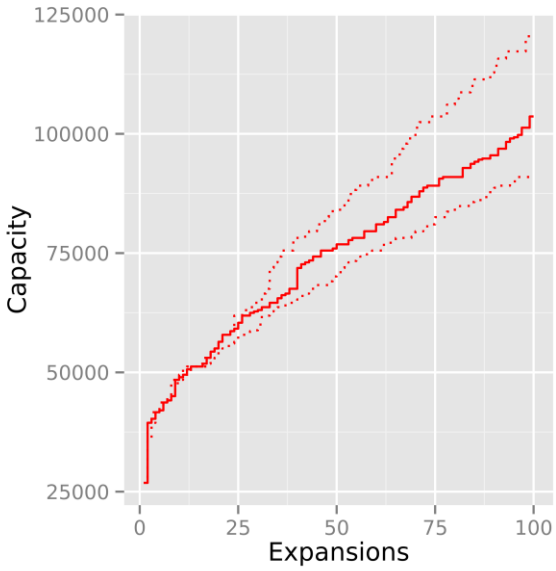


Fig. 1. The maximum number of subscribers as a function of the number of cell expansions for the expansion factors 2, 3/2 (solid red line), and 4/3 respectively.

3.1 Cell Expansions and Tetris Optimization

One way of combining Tetris optimization and cell expansion is to first do Tetris optimization, and then do cell expansion with the user mix obtained after the Tetris optimization. We evaluated this approach by first doing the Tetris optimization, thus obtaining the x-values (scaling factors) presented above. We called these scaling factors **x**', i.e., $x_1' = 0$, $x_2' = 0.13$, $x_3' = 0$, $x_4' = 1.45$, $x_5' = 4.85$, $x_6' = 0.92$. We then added the restrictions that $x_1 = x_3 = 0$, and $x_2/x_2' = x_4/x_4' = x_5/x_5' = x_6/x_6'$ to our linear programming problem. These new restrictions, preserve the user mix to the one obtained after Tetris optimization. We then do cell expansion in the same way as above, i.e., by identifying the cells associated with the active restrictions and multiplying the capacity of these cells with a factor 3/2 (or 2, or 4/3).

Fig. 2, shows the result of first doing Tetris optimization and then cell expansion, for the first 100 cell expansions. For the expansion factors 2 and 4/3 we see that the difference in terms of the maximum number subscribers increases when the number of cell expansions increases. The reason for this is that for the lower expansion factors, more cells need to be expanded multiple times.

4 Next steps

We plan to continue to explore the potential of the Tetris optimization approach. Currently, we are using geodemographic user segment (see above). In the next phase we will investigate user segments (user clusters) based on the subscribers' mobility patterns. We think that the gain doing Tetris optimization based on mobility based subscriber clusters will be even greater than the gain of doing Tetris optimization for segments based on geodemographic data (which is what we have done so far). As a part of our future work we will also evaluate how different clustering algorithms affects the gain of doing Tetris optimization.

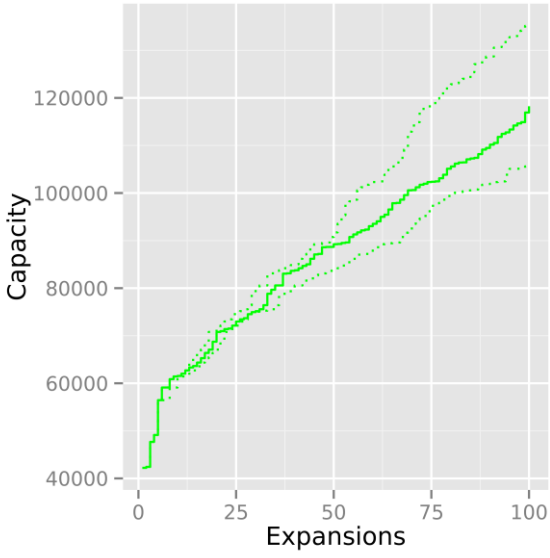


Fig. 2. The maximum number of subscribers as a function of the number of cell expansions after initial Tetris optimization for the expansion factors 2, 3/2 (solid green line), and 4/3 respectively.

References

[1] Abichar, Z., Kamal, A. E., & Chang, J. M. (2010, April). Planning of relay station locations in IEEE 802.16 (WiMAX) networks. In *2010 IEEE Wireless Communication and Networking Conference* (pp. 1-6). IEEE.

[2] Amaldi, E., Belotti, P., Capone, A., & Malucelli, F. (2006). Optimizing base station location and configuration in UMTS networks. *Annals of Operations Research*, *146*(1), 135-151.

[3] Amaldi, E., Capone, A., & Malucelli, F. (2008). Radio planning and coverage optimization of 3G cellular networks. *Wireless Networks*, *14*(4), 435-447.

[4] Athanasiadou, G. E., Tsoulos, G. V., & Zarbouti, D. (2015, May). A combinatorial algorithm for base-station location optimization for LTE mixed-cell MIMO wireless systems. In *2015 9th European Conference on Antennas and Propagation (EuCAP)* (pp. 1-5). IEEE.

[5] Bhushan, N., Li, J., Malladi, D., Gilmore, R., Brenner, D., Damnjanovic, A., ... & Geirhofer, S. (2014). Network densification: the dominant theme for wireless evolution into 5G. *IEEE Communications Magazine*, *52*(2), 82-89.

[6] Debenham, J., Clarke, G., & Stillwell, J. (2003). Extending geodemographic classification: a new regional prototype. *Environment and Planning A*, *35*(6), 1025-1050.

[7] Optimization, Gurobi, 2012. Inc.: Gurobi optimizer reference manual.

[8] Grubesic, T. H. (2004). The geodemographic correlates of broadband access and availability in the United States. *Telematics and Informatics*, *21*(4), 335-358.

[9] González-Brevis, P., Gondzio, J., Fan, Y., Poor, H. V., Thompson, J., Krikidis, I., & Chung, P. J. (2011, May). Base station location optimization for minimal energy consumption in wireless networks. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd* (pp. 1-5). IEEE.

[10] Haenlein, M., & Kaplan, A. M. (2009). Unprofitable customers and their management. *Business Horizons*, *52*(1), 89-97.

[11] Hurley, S. (2002). Planning effective cellular mobile radio networks. *IEEE Transactions on Vehicular Technology*, *51*(2), 243-253.

[12] Ibbetson, L. J., & Lopes, L. B. (1997, May). An automatic base site placement algorithm. In *Vehicular Technology Conference, 1997, IEEE 47th* (Vol. 2, pp. 760-764). IEEE.

[13] Islam, M. H., Dziong, Z., Sohraby, K., Daneshmand, M. F., & Jana, R. (2012, February). Capacity-optimal relay and base station placement in wireless networks. In *The International Conference on Information Network 2012* (pp. 358-363). IEEE.

[14] Levy, J. (2006) How to market better health -- diabetes. A Dr. Foster Community Health Workbook. London: Dr. Foster.

[15] Mathar, R., & Niessen, T. (2000). Optimum positioning of base stations for cellular radio networks. *Wireless Networks*, *6*(6), 421-428.

[16] Molina, A., Athanasiadou, G. E., & Nix, A. R. (1999, July). The automatic location of base-stations for optimised cellular coverage: A new combinatorial approach. In *Vehicular Technology Conference, 1999 IEEE 49th* (Vol. 1, pp. 606-610). IEEE.

[17] Naboulsi, D., Fiore, M., Ribot, S., & Stanica, R. (2015). Large-scale mobile traffic analysis: a survey. *IEEE Communications Surveys & Tutorials*, *18*(1), 124-161.

[18] Richter, F., & Fettweis, G. (2012, May). Base station placement based on force fields. In *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th* (pp. 1-5). IEEE.

[19] Siqueira, G. L., Vasquez, E. A., Gomes, R. A., Sampaio, C. B., & Socorro, M. A. (1997, May). Optimization of base station antenna position based on propagation measurements on dense urban microcells. In *Vehicular Technology Conference, 1997, IEEE 47th* (Vol. 2, pp. 1133-1137). IEEE.

[20] Tutschku, K., & Tran-Gia, P. (1998). Spatial traffic estimation and characterization for mobile communication network design. *IEEE Journal on selected areas in communications*, *16*(5), 804-811.

[21] Tutschku, K. (1998, April). Demand-based radio network planning of cellular mobile communication systems. In *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 3, pp. 1054-1061). IEEE.

[22] Valavanis, I. K., Athanasiadou, G., Zarbouti, D., & Tsoulos, G. V. (2014, May). Base-Station Location Optimization for LTE Systems with Genetic Algorithms. In *European Wireless 2014; 20th European Wireless Conference; Proceedings of* (pp. 1-6). VDE.

[23] Wang, S., Zhao, W., & Wang, C. (2015). Budgeted cell planning for cellular networks with small cells. *IEEE Transactions on Vehicular Technology*, *64*(10), 4797-4806.

[24] Webber, R., & Butler, T. (2007). Classifying pupils by where they live: how well does this predict variations in their GCSE results?. *Urban Studies*, *44*(7), 1229-1253.

[25] Webber, R. (2009). Response toThe Coming Crisis of Empirical Sociology': An Outline of the Research Potential of Administrative and Transactional Data. *Sociology*, *43*(1), 169-178.

[26] Yang, J., Aydin, M. E., Zhang, J., & Maple, C. (2007). UMTS base station location planning: a mathematical model and heuristic optimisation algorithms. *IET communications*, *1*(5), 1007-1014.

[27] Yu, Y., Murphy, S., & Murphy, L. (2008, January). Planning base station and relay station locations in IEEE 802.16 j multi-hop relay networks. In *2008 5th IEEE Consumer Communications and Networking Conference* (pp. 922-926). IEEE.

Large Scale Graph Exploration

Discovering notable characteristics among nodes in knowledge graphs

Davide Mottin[1] and Emmanuel Müller[1]

[1]Hasso Plattner Institute - Prof. Dr. Helmert-Str. 2-3, 14482 Potsdam
{davide.mottin,emmanuel.mueller}@hpi.de

Abstract

We are witnessing an unprecedented increase in the adoption of graphs for different applications, from biological to social. However, while their size and complexity keep growing, automatic graph exploration methods are limited to a few specialized techniques. Consider the case in which a user wants to compare nodes in the graph and see how they differ from their similars. This is the case of a student searching for differences in two or more presidents, or a biologist looking at two microorganisms. In this work, we propose a novel formulation to discover what we call notable characteristics *given an initial set of nodes. We propose a solid probabilistic approach that first retrieves nodes that are similar to the seed provided by the user, and then exploits distributional properties to understand whether a particular attribute is interesting or not. We experimentally evaluate the effectiveness of our approach and show that we are able to discover notable characteristics that are indeed interesting and relevant for the user.*

1 Introduction

Consider the case in which a user wants to compare nodes in the graph and see how they differ from their similars. This is the case of a student searching for differences in two or more presidents, or a biologist looking at two microorganisms. Traditionally, the user would look at the nodes, their relationships and attributes and select those that are more interesting. However, this painful approach requires a long time and would lead to scarce results or errors. In this work, we propose a novel formulation to discover what we call *notable characteristics* given an initial set of nodes. While the traditional comparison of nodes by means of node similarity provides only a score with no explanation, we go one step further. We propose a solid probabilistic approach that first retrieves nodes that are similar to the seed provided by the user, and then exploits distributional properties to understand whether a particular attribute is interesting or not. We experimentally evaluate the effectiveness of our approach and show that we are able to discover notable characteristics that are indeed interesting and relevant for the user.

A knowledge graph is a directed graph in which nodes and edges have labels or types. They are also known as information networks [4, 6] or simply labeled graphs. We are given a set $\mathcal{A}$ of node labels and a set $\mathcal{L}$ of edge labels. The term label and type are used interchangeably.

Definition 1 (Knowledge graph)**.** *A knowledge graph is a quadruple $G : \langle \mathcal{V}, \mathcal{E}, \phi, \psi \rangle$, where $\mathcal{V}$ is a set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, $\phi : \mathcal{V} \mapsto \mathcal{A}$, $\psi : \mathcal{E} \mapsto \mathcal{L}$ are node and edge labeling functions, respectively.*

Recall that we are interested in discovering *notable characteristics* of the entities mentioned in a set of input nodes in relation to their similars. This intuitive definition entails two questions: (1) what is the set of similars? (2) what are the notable characteristics?
The set of input nodes is referred to as *seed set* (seeds in short). Formally, given a knowledge graph $G : \langle \mathcal{V}, \mathcal{E}, \phi, \psi \rangle$ the set of seed nodes is any set $S \subseteq \mathcal{V}$. The seed set is manually provide by the user and therefore considered reasonably small (i.e., ≤ 10 elements). Given any set of seed nodes, we need to define a set of similars or *context nodes*. We assume the existence of a similarity function $\sigma : \mathcal{V} \times 2^{\mathcal{V}} \mapsto \mathbb{R}$ that assigns a high score to nodes that are similar to those in the seed set and low otherwise. Then, the context are the top-k most similar nodes.

Definition 2 (Context set)**.** *Given a knowledge graph $G : \langle \mathcal{V}, \mathcal{E}, \phi, \psi \rangle$, a seed set $S \subseteq \mathcal{V}$, a similarity function $\sigma : \mathcal{V} \times 2^{\mathcal{V}} \mapsto \mathbb{R}$, and a parameter k, the* context set *(or simply context) is a set $C \subseteq \mathcal{V}$ such that*

$S \subseteq C, |C| = k$, *and for each* $n_c \in C \wedge n \in \mathcal{V} \setminus C, \sigma(n, S) \leq \sigma(n_c, S)$.

The second question concerns the notable characteristics. The characteristics are attributes or relationships of a specific node since they implicitly represent a signature of the node itself. As before, we assume the existence of a generic discrimination function, whose role is to return a score whether a specific characteristic is discriminative or unexpected comparing two set of nodes. Formally, in the knowledge graph G, a discrimination function $\delta : \mathcal{L} \times 2^{\mathcal{V}} \times 2^{\mathcal{V}} \mapsto \mathbb{R}_0^+$ assigns a discrimination value or 0 if the value is not discriminative. We are now ready to define a notable characteristic.

Definition 3 (Notable characteristic)**.** *Given a knowledge graph* $G : \langle \mathcal{V}, \mathcal{E}, \phi, \psi \rangle$*, a seed set* $S \subseteq \mathcal{V}$*, a context* $C \subseteq \mathcal{V}$*, and a discrimination function* $\delta : \mathcal{L} \times 2^{\mathcal{V}} \times 2^{\mathcal{V}} \mapsto \mathbb{R}_0^+$ *a* notable characteristic *is a relationship* $l \in \mathcal{L}|_C$ *such that* $\delta(l, S, C) \neq 0$.

The notation $\mathcal{L}|_C$ denotes the set of edge labels restricted to those that are found in the edges directly connected to C, i.e., $\mathcal{L}|_C = \{l \mid \exists x \in C, y \in \mathcal{V}$ s.t. $(x, y) \in \mathcal{E} \wedge \psi(x, y) = l\}$.
The general problem we aim to solve is efficiently returning the notable characteristics, given a seed set, a similarity function and a discrimination function.

Problem 1 (Finding notable characteristics)**.** *Given a knowledge graph* $G : \langle \mathcal{V}, \mathcal{E}, \phi, \psi \rangle$*, a seed set* $S \subseteq V$*, a similarity function* $\sigma : \mathcal{V} \times 2^{\mathcal{V}} \to \mathbb{R}$ *and a discrimination function* $\delta : \mathcal{L} \times 2^{\mathcal{V}} \times 2^{\mathcal{V}} \mapsto \mathbb{R}_0^+$*, find the set of notable characteristics.*

The problem entails the definition of suitable functions σ and δ that are able to retrieve and compare nodes.

2 Approach

We model the discrimination function in probabilistic terms, in order to better deal with noisy settings and uncertainty. Therefore, we assume that a characteristic is interesting if its distribution in the seed set deviates from the one in the context set. In other words, the context represents the expected behavior of the population while the seed is the hypothesis to be tested.
Given the seed set S, the first step requires the definition of a similarity function σ to retrieve a set of context nodes. Although many notions of similarity functions have been developed, such as structural equivalence [5] and SimRank [2], none seems suitable to our case. Existing similarity measures are either based on restricted neighborhoods of the nodes [5], or they disregard edge and node labels [2] We propose an algorithm that takes into account the kind of connections between pairs of nodes and combines the advantages of random walk and metapath approaches.

In the traditional random walk model, a random walker chooses one of the outgoing edges from a node with uniform probability. Instead of uniform probability, we favor choices which are more informative in terms of edge label. Intuitively, an edge label is informative if it has low frequency. This intuition is supported by information theoretic notions, such as tf-idf and has been successfully used in graphs as well [7]. As a shorthand notation, we define $\mathcal{E}_l$ as the set of edges having label $l \in \mathcal{L}$, i.e., $\mathcal{E}_l = \{(i, j) \in \mathcal{E} | i, j \in \mathcal{V}, \psi(i, j) = l\}$. The frequency of a label l is the fraction of l-labeled edges with respect to the total number of edges. We then define the weighted adjacency matrix as a $|V| \times |V|$ square matrix, where the value A_{ij} between node i and j is defined as

$$A_{ij} = \begin{cases} 1 - |\mathcal{E}_l|/|\mathcal{E}| & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The Personalized PageRank is defined as the vector

$$\mathbf{p} = c * \tilde{A} * \mathbf{p} + (1 - c) * \mathbf{v}, \quad (2)$$

where $\tilde{A}_{ij} = A_{ji} / \sum_k A_{jk}$, c is the damping factor, and $\mathbf{v}$ is vector called personalization vector. In our experiments the damping factor is 0.8, in line with previous works. We compute the PageRank starting from each node in the seed set to retrieve the k nodes with the highest score. This is done by setting $\mathbf{v}_n = 1$ for each $n \in S$, individually. We refer to this baseline as RANDOMWALK.
However, the RANDOMWALK baseline disregards common connections between the seed nodes. This is an important information, since the tf-idf approach does not consider the user's similarity notion implicitly contained in the seed set. To this end, we adopt the notion of metapath from [8] which generalizes the concept of path. A metapath for a path $\langle n_1, ..., n_t \rangle, n_i \in \mathcal{V}, 1 \leq i \leq t$ is defined as the sequence $\langle \phi(n_1), \psi(n_1, n_2), ..., \psi(n_{t-1}, n_t), \phi(n_t) \rangle$ that alternates node and edge labels along the path. We mine metapaths running PathMining [3] from the seed nodes. Differently from the original PathMining, we start from several nodes with uniform probability and consider only edge types. Our algorithm stops the exploration when another seed node is encountered. The metapaths and the counts for each path are separately stored to compute the similarity score.
Once the metapaths are computed, we compute a score for each node based on the probability that some metapath starting from a seed node ends in this node. Given the set of metapaths M obtained with our modified PathMining, we denote as $n \overset{m}{\rightsquigarrow} n'$ any path from n to $n' \in \mathcal{V} \setminus S$ matching metapath $m \in M$. Therefore, the score of a node $n' \in \mathcal{V} \setminus S$ with respect to any seed node $n \in S$ is

$$\sigma(n', S) = \sum_{m \in M, n \in S} \frac{|\{n \overset{m}{\rightsquigarrow} n'\}|}{|\{n \overset{m}{\rightsquigarrow} n'' | n'' \in \mathcal{V} \setminus S\}|} \Pr(m)$$

$\Pr(m)$ is the probability of choosing metapath m, which is the relative count computed previously divided by the sum of the counts of all metapaths M. Intuitively, σ gives a higher score to nodes that are reachable through frequent metapaths connecting the seed nodes or connected through many of these metapaths. This means that nodes that are reached from infrequent metapaths will have a low score. Once we have computed the score for each node we return the k nodes with the highest score as our context.

Assume we have computed the distribution of values for each characteristic (i.e., edge label) for both seed nodes and context nodes found with the previous method. Intuitively, for each characteristic, the distribution of the context represents the expected, or normal behavior, the one to compare with. Therefore, the seed set becomes the hypothesis to be evaluated against the "true" distribution of the context.

Formally, for each characteristic $l \in \mathcal{L}$, we consider two distributions in order to evaluate its notability. The first evaluates the number of occurrences of a distinct node label instance connected through a specific edge label (e.g., bornIn, California). This expresses information about the actual attribute values of the nodes and can be used to identify cases where different attribute values are relevant. For instance, most people in the seeds are born in the U.S., while those in the context are equally born in the U.S. and Europe. We refer to these distributions as *instance distributions*.

$$Inst_s(l, C, S) = (x_1, x_2, ..., x_m),$$
$$Inst_c(l, C) = (y_1, y_2, ..., y_m)$$

where x_i and y_i are the number of occurrences of node i at the end of an edge labeled l from a node in S and C, respectively. Note that both vectors have the same size, so x_i is zero if i appears only in the context. Similarly, a second distribution computes aggregates over the number of occurrences of a specific edge type in the context. This expresses information about the existence and cardinality of an attribute and can be used to identify cases where attribute cardinality is relevant. For instance, people in the seed all have a single child while in the context most have two children. Such cases cannot evidently be modeled as instance distributions that take into account distinct values. We refer to these distributions as *cardinality distributions*.

$$Card_s(l, C, S) = (x_1, x_2, ..., x_m),$$
$$Card_c(l, C) = (y_1, y_2, ..., y_m)$$

where x_i and y_i are the number of times a node in S and C respectively has i edges labeled l.

Both distributions can be built by iterating through the nodes in each set and counting the respective occurrences. For a given $l \in \mathcal{L}$, this results in two scores δ_{Inst} and δ_{Card} for instance and count distributions. The final score δ is a maximum aggregation score between δ_{Inst} and δ_{Card}.

$$\delta(l, C, S) = \max(\delta_{Inst}(l, C, S), \delta_{Card}(l, C, S)) \tag{3}$$

Many measures have been proposed in statistics to compare two distributions. However, most of them draw specific assumptions, such as a minimum number of samples or non-zero probabilities, that are not fulfilled in our case. In particular, $Inst$ and $Card$ have no natural ordering and no distance-function between the values. Additionally, we compare a m-sized distribution over our context, where m is the size of the context, to a much smaller distribution over the seed-nodes. This leads to many zero values in the seed-distribution.

We resorted to a more natural multinomial test that better expresses the relationship between our distributions. The multinomial test assumes that a set of observations is drawn from a known multinomial distribution. Therefore, assuming the context to be Multinomial distributed the observations are the values found in the seed set. If the values observed in the seed sets are drawn from the Multinomial, than the hypothesis cannot be rejected and the characteristic is marked as non-notable. On the other hand, if the test succeeds, then the two distributions are significantly different and the characteristic is notable.

Assume we have a random variable $X_{N,\pi} \sim Mult(N, \pi)$, with parameters N and π. We normalize $Inst_c$ and $Card_c$ to express a probability distribution $\pi = normalize(y) = (\pi_1, \pi_2, ..., \pi_k)$. For a given outcome $x = (x_1, x_2, ..., x_k)$, the probability, under the hypothesis of equality between context and sample, is

$$\Pr(\mathrm{X_{N,\pi}} = \mathrm{x}) = N! \prod_{i=1}^{k} \frac{\pi_i^{x_i}}{x_i!},$$

where $N = \sum x_i$. In an exact multinomial test, the significance probability is

$$\Pr_s(X_{N,\pi} = x) = \sum_{\substack{y:\Pr(X_{N,\pi}=y) \leq \\ \Pr(X_{N,\pi}=x)}} \Pr(X_{N,\pi} = y)$$

$\Pr_s(\pi, x)$ is the probability of x or any equally or less likely outcome being drawn from the probability distribution. In case of large N, the exact test is impractical, we therefore perform a Montecarlo sampling to approximate the final result. A difference in distributions is considered significant, if the hypothesis is rejected with probability $p > 0.95$.

3 Experimental eveluation

Datasets: We perform experiments on two datasets: YAGO and LinkedMDB.

• YAGO [1] is a large knowledge graph based on Wikipedia, Wordnet and Geonames. We downloaded

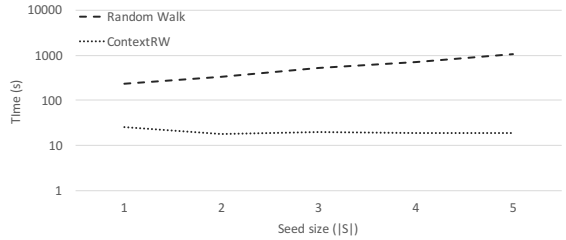


Figure 1: Average time (s) vs seed set size ($|S|$) comparison in YAGO dataset.

YAGO 2.5[1] core facts in April 2016. It consists of 3.3M nodes and 27M edges, with 366K node types and 38 edge labels, including a type-hierarchy for node types. We represented each node attribute as an edge, having the attribute value as node label.

• LinkedMDB is a knowledge graph for the movie domain, extracted from the Internet Movie Database (IMDB). We downloaded a snapshot of LinkedMDB[2] in June 2016. It consists of 739K nodes and 1.6M edges of 18 types.

Experimental Setup: We implemented our solution in Java 1.8, and ran the experiments on a FutureSoc Lab machine with a quad-core Intel CPU 1.7 GHz and 64GB RAM. All the datasets are loaded into Apache Jena triple store to perform quick traversals on the graph without loading it into main memory. The impemented algorithms are the following.

• RANDOMWALK: A baseline algorithm for context selection based on Personalized PageRank. Instead of the matrix multiplication we used the more scalable power iteration method. We set the number of iterations to 10 and the damping factor $c = 0.8$.

• CONTEXTRW: This is our algorithm that includes PathMining to mine the metapaths, the weighted random walk constrained to the metapaths found by PathMining, and the final score. We ran PathMining 1M times to retrieve the relevant metapaths.

• FINDNC: This is the final algorithm that combines CONTEXTRW and the method previously described. For the multinomial test, we used a statistic package written in R.

Summary of the experiments: We evaluate our algorithms effectiveness by comparing the found context to a ground truth obtained through a user survey. Our context selection returns a better context compared to the baseline quicker. Moreover, our algorithm performs better as the seed set increases. The returned notable characteristics indeed represent interesting undisclosed facts in the seeds.

Seed size ($|S|$). The seed size $|S|$ affects both time and quality. We compare the total runtime of each method varying the seed size ($|S|$). Figure 1 shows the time to compute the context for CONTEXTRW and the baseline RANDOMWALK. We note that the RANDOMWALK algorithm is on average up to two orders of magnitude slower than CONTEXTRW, for $|S| = 5$. Moreover, while CONTEXTRW is faster with larger seed sets, a random walk approach tends to become slower. This is an expected behavior in CONTEXTRW, since the chances to end the exploration in a seed node is larger as the seed set size increases. Furthermore, we are able to return results in less than 20s.

[1] http://resources.mpi-inf.mpg.de/yago-naga/yago2.5/yago2s_tsv.7z

[2] https://datahub.io/dataset/linkedmdb

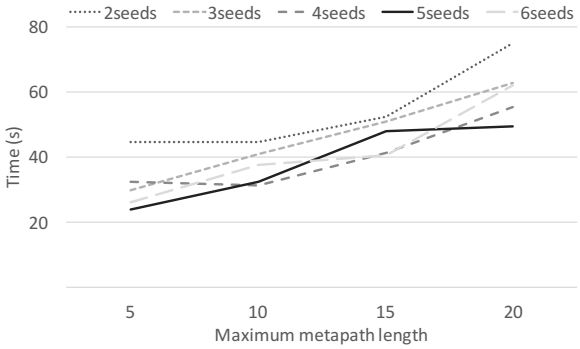


Figure 2: Time (s) vs metapath length with different seed set sizes ($|S|$)

	Number of paths ($|M|$)			
$|C|$	5	10	15	20
50	0.15	0.16	0.13	0.15
100	0.22	0.21	0.21	0.21
150	0.22	0.23	0.23	0.23
200	0.22	0.22	0.22	0.22

Table 1: F_1 score as a function of the number of paths $|M|$ and the size of the context $|C|$ for CONTEXTRW algorithm.

Seeds		$\max F_1$	$|C|$
2	YAGO	0.23	23
	LinkedMDB	0.30	101
3	YAGO	0.2	107
	LinkedMDB	0.25	122
4	YAGO	0.19	130
	LinkedMDB	0.24	124
5	YAGO	0.25	162
	LinkedMDB	0.26	198
6	YAGO	0.22	285
	LinkedMDB	0.25	139

Table 2: Comparing the performance of CONTEXTRW on YAGO and LinkedMDB in the *actors* domain.

Table 2 reports the maximum F_1 score at increasing $|S|$, comparing YAGO and LinkedMDB datasets within the *actors* domain using the CONTEXTRW algorithm. While we could not evaluate for the *politicians* domain because the knowledge is not included in the LinkedMDB dataset, the results for *movie contributors* are mostly comparable and omitted for

brevity. Unsurprisingly, CONTEXTRW performs better in LinkedMDB due to the specificity of the dataset. However, the overall maximum increasing in F_1 is not larger than 0.7. This supports the claim that CONTEXTRW is able to capture domain specific knowledge even in more general datasets, exploiting the characteristics of the graph and the metapaths.

Number of paths ($|M|$). The CONTEXTRW algorithm depends on the number of paths. Table 1 shows the F_1 score in relation to the context size and the number of paths. The number of paths does not affect the score; however, as shown in Figure 2 the time increases as the length of the metapaths (and also the number, not reported) increases. Therefore, a reasonable choice for the number of metapaths $|M|$ and maximum length is 5.

3.1 Distribution Comparison

We evaluate the performance of the FINDNC algorithm in terms of quality.

Metrics comparison. We first evaluate the results comparing the characteristics found by FINDNC with those found by KL-divergence, and EMD that allow distribution comparison. We asked three judges to provide a score to the characteristics of a small set of examples. We then aggregated the individual judgments and compared the ranking with the one obtained by the three methods. The minimum number of switches needed to transform one ranking to the other was used as a metric. We found that FINDNC required 2 changes, while KL-divergence and EMD required 4 and 5, respectively, supporting the choice of the multinomial test as a measure of quality.

Test cases. Evaluating the quality of the results objectively is impractical given the subjectivity of the notion of notable characteristics. We therefore resorted to an anecdotal analysis of two test cases to show that FINDNC finds results that are more interesting than the one retrieved by the baseline RANDOMWALK when equipped with the multinomial test. We refer to RANDOMWALK with multinomial test as RWMULT.
One test case includes the scenario with the best F_1 score for the context construction, that has $S = \{George\ Clooney,\ Brad\ Pitt,\ Leonardo\ DiCaprio,\ Scarlett\ Johansson,\ Johnny\ Depp\}$ as the seed set. We selected the top 100 nodes as the context. The distribution comparison with multinomial test identified multiple edge labels, for which we provide a visual analysis of the findings.

4 FutureSoc-Lab Resources

In this project we used a single server quad-core machine and loaded the graph into a graph database. The results show that we are able to scale to real world graphs and return answers in a few seconds. The use of FutureSoc-Lab resources helped us in performing all the experiments and achieving the initial goalds.

5 Conclusions and future work

In this project, we studied the problem of discovering notable characteristics given a set of seed nodes in a knowledge graph. A notable characteristic is a special property in the seed nodes that makes them different from their similars. Our problem is twofold: We first find a context set that represents the nodes similar to the seeds; we then identify the notable characteristics with a novel probabilistic framework. We devise an algorithm for context selection based on random walk and metapath discovery and prove its effectiveness and efficiency with real data and user generated ground truth. In order to find the notable characteristics, we propose a probabilistic notion that first computes distributions for each edge label and subsequently performs a multinomial test to mark the characteristics that deviate from the expected behavior. We show different test cases to demonstrate the applicability and the effectiveness of our approach in real dataset.
As future work we plan to expand the notion of notable characteristics to incorporate more complex patterns. We also intend to explore correlations between attributes as well as graph structures and incorporate results into the model.

References

[1] J. Biega, E. Kuzey, and F. M. Suchanek. Inside yago2s: A transparent information extraction architecture. In *WWW*, pages 325–328, 2013.

[2] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *KDD*, pages 538–543, 2002.

[3] S. Lee, S. Lee, and B.-H. Park. Pathmining: A path-based user profiling algorithm for heterogeneous graph-based recommender systems. In *FLAIRS Conference*, pages 519–523, 2015.

[4] S. Lee, S. Park, M. Kahng, and S.-g. Lee. Pathrank: a novel node ranking measure on a heterogeneous graph for recommender systems. In *CIKM*, pages 1637–1641, 2012.

[5] F. Lorrain and H. C. White. Structural equivalence of individuals in social networks. *The Journal of mathematical sociology*, 1(1):49–80, 1971.

[6] C. Meng, R. Cheng, S. Maniu, P. Senellart, and W. Zhang. Discovering meta-paths in large heterogeneous information networks. In *WWW*, pages 754–764, 2015.

[7] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas. Exemplar queries: Give me an example of what you need. *PVLDB*, 7(5):365–376, 2014.

[8] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

Small Road Network Alterations to Measure Effect Ranges and to Identify Super-Sensitive and Braess Roads

Jordan Ivanchev
TUM Create
1 Create Way
Singapore 138602
jordan.ivanchev@tum-create.edu.sg

Daniel Zehe
TUM Create
1 Create Way
Singapore 138602
daniel.zehe@tum-create.edu.sg

Suraj Nair
TUM Create
1 Create Way
Singapore 138602
suraj.nair@tum-create.edu.sg

Alois Knoll
Technical University Munich
Arcisstraße 21
80333 München, Germany
knoll@in.tum.de

Abstract

This paper deals with measuring the changes of traffic conditions as a result of small alterations introduced to the road network, namely addition and removal of a lane from a single road segment. By measuring the change of total travel time under user equilibrium in the system we are able to determine road segments, which when changed significantly affect the traffic conditions. Those super-sensitive *road segments can then be utilized for steering traffic conditions on a global level. Identification of such segments, however, is a non-trivial task, requiring significant computational power for search space exploration in a realistic large city scenario. The city of Singapore has been used as a case study with realistic traffic demand and a complete road network. We have parallelized the traffic assignment problem algorithm, which enables us to use high performance computation techniques in order to maximize our exploration capabilities. Furthermore, the generated results have been used to confirm the existence of the Braess paradox roads in the examined system and to demonstrate the long spatial range of the effects resulting from the introduced small road network changes.*

1 Introduction

Transportation networks in cities are complex systems wherein, small changes applied at sensitive locations in transportation networks can have a disproportionately large impact on the city dynamics. This means that well-planned interventions can be used to efficiently steer traffic conditions.
Identifying such locations and quantifying their *super-sensitivity* is a non-trivial task. This paper deals with finding such places in a real-world scenario of a large-city road network with calibrated traffic data. Our approach consists of examining a predetermined set of interesting locations in a systematic manner. The examination of a suitable location consists of computing the traffic equilibrium assignment before and after a small alteration of the traffic infrastructure is done and measuring the degree of effect on a global scale.
Similar studies for identification of important or sensitive locations have been performed in [9, 8, 10]. Furthermore, in [11] it has been demonstrated that if small changes were applied to certain locations of the road network, the total travel time of the commuting population can be significantly reduced, especially during rush hour. The main difference of the undertaken approach in this paper is the traffic assignment method, which produces more realistic results at the price of a significant increase of required computational power.
"Critical" intersections as they are referred to in literature are the locations, which attract the most interest in the analysis of a road network. The measure of criticality is considered to be correlated with the volume of passing vehicles as discussed in [7] and [2], where traffic management strategies are also proposed. There have also been efforts to define a critical traffic volume as in [6], which is used to make the decision whether to take on active traffic control in the sense of traffic lights in order to optimize traffic conditions on the intersection.
It must be noted that typically transportation researchers optimize locally areas or even intersections in order to maximize a certain throughput. Changes at some locations can, however, induce significant differences at other distant roads, which are not considered in the optimization problem. Our study aims at ob-

serving the effects of a local infrastructural change on the whole system in order to either validate or advice against such local optimization approaches.
The reason why it is possible for a small change to have effects on distant locations lies in the complexity of the system, specifically in the traffic assignment of vehicles. A reasonable approximation of the way commuters choose their routes is provided by the user equilibrium (UE) traffic assignment [3]. Such UE occurs when all commuters have computed their shortest routes in terms of time, having perfect information about the system and the other commuters' choices. This is analogous to the Nash equilibrium also known as the Wardrop's equilibrium, where no user would voluntarily change his/her route choice. In the event of a small change in infrastructure, a group of users might have incentive to switch their routes, which in turn might lead to an avalanche effect of changing routes within the system. This state of equilibrium has been accepted to represent real traffic conditions because it is believed that the commuting population slowly converges to it. It is not assumed that everyone has perfect information, however, by incrementally trying routes the users eventually reach this equilibrium state. For more information of modelling techniques for traffic assignment and route choice theory the reader can refer to [14].
It must be noted that such a traffic assignment does not lead to a minimum overall travel time, because of the lack of coordination between the commuters. In the presence of a centralized routing system, which distributes traffic evenly a system optimum (SO) traffic assignment will be reached. The ratio between system performance for SO and UE is called "Price of Anarchy" [12], representing the consequences of selfish routing of commuters.
One phenomenon, which exists under UE but is impossible under SO traffic assignment is the Braess paradox [4]. It states that if a road is added to the system it might decrease the performance of the system and alternatively if a road is removed the system might benefit from this action. It has been shown in [11] that the paradox exists in a realistic large city environment under shortest path routing. This study will check for the existence of the paradox under UE traffic assignment.

2 Methods and Experiment Description

2.1 Macroscopic Simulation

The three main elements needed to enable our macroscopic simulation are the road network graph, the origin - destination pairs of the population and the routes that the commuters choose.
A road network of the simulated system is available to us, including speed limits, number of lanes on every road segment, and connectivity between the edges of the graph. A realistic number of drivers ($375,000$) are generated by sampling their origins and destinations from a survey data set of real start and end trip points. After this UE traffic assignment is performed to compute the routes of the drivers as described in [13].
After the routes are computed the number of drivers passing through every road segment during the simulation period T_S can be extracted. Knowing the flow F_i, length l_i, free flow speed v_i^f, number of lanes w_i and the coefficients α_i and β_i, which are calibrated for road i (see a detailed description in [11]), we can estimate the average traverse time t_i along every road segment using the Bureau of Public Roads (BPR) function [5]:

$$t_i = \frac{l_i}{v_i^f}\left(1 + \alpha_i\left(\frac{F_i}{2000 w_i T_S}\right)^{\beta_i}\right) \qquad (1)$$

Our case study examines the city of Singapore with population of 5.4 million people and around 1 million registered vehicles including taxis, delivery vans and public transportation vehicles [1]. It is an island city, thus making the examined system relatively closed. The road network graph comprises of $240,000$ edges and $160,000$ nodes.
Two datasets have been used in order to calibrate and validate the macroscopic simulation. The first one is the Household Interview Travel Survey (HITS) conducted in 2012 in the city of Singapore, which provides information about the traffic patterns of commuters. The second data set represents GPS traces of a $20,000$ vehicles fleet for the duration of one month.

2.2 Experiment Description

The experiment performed in this paper consists of going through pre-selected links of interest and measuring the sensitivity of the system to their capacity. Let's assume that the computed total travel time of the system is T. Let link i have w_i lanes. The first step is to set the number of lanes to $w_i - 1$. New UE traffic assignment is performed and the total travel time T_i^{-1} is computed. The second step is to set the number of lanes to $w_i + 1$ and assign the traffic once again to compute T_i^{+1}. By computing the difference between the computed travel times, the sensitivity of the system to the capacity of this road segment can be estimated. In a more abstract way, we are actually computing the partial derivative of the total travel time under UE with respect to the selected link. This procedure was done for roughly $2,500$ road segments, which were chosen based on both their high throughput at UE or congestion factor.

2.3 Resources Utilization

For the experiment we used 13 nodes of the 25 total nodes available on the HPI Future SOC Lab 1000 core cluster. Each node of the cluster is equipped with 4

Intel Xeon E7- 4870 processors as well as 1024 Gb of main memory. Since hyper-threading was enabled by the cluster operators, the number of threads that could be used concurrently was 80. In order to improve the utilization of each node, the simulation software was not using all available computing resources in a single application on each node, but was rather dividing the workload into 4 applications with different input data. This was done to better exploit the shared-memory parallelization of the routing algorithm within the experiment application.
Hence, the input data was split into 52 unique portions. The result of distributing the workload to 4 distinct runs on each node improved the overall calculation time for each batch (of 200 routes) to around 1 second, while using 20 threads. The memory requirement of the experiment application was low, since only the routing network had to be kept in memory for each of the 20 threads of an application as well as the agent information (e.g., routes). This summed up to be 30 Gb per application instance. Since every application worked on their distinct data set of input data, there was no synchronization or communication between the 53 instances necessary. The output files generated were 8.4Mb in size and since we have tried out around 2500 road segments where we added and removed (if possible) lanes, the total disk space required was around 38Gb. A total of 15 billion route computations on the 240,000 edge graph were executed in order to solve the UE traffic assignment problem for the various road network combinations.

3 Results

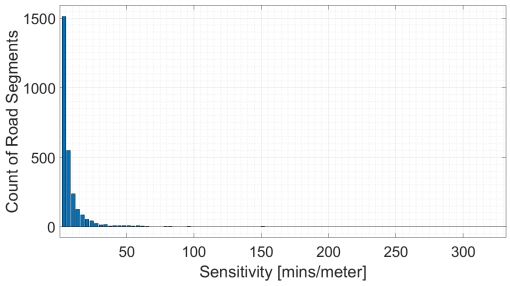
The results acquired from the study can be used in three main directions. First, by measuring the sensitivity of the system to capacity changes in certain road segments, we can identify *super-sensitive* locations, which can be used as steering tools for traffic management. Second, based on the results, we can identify Braess road segments. Such segments exhibit paradoxical behavior, where either adding capacity to them worsens the traffic conditions or removing capacity from them improves traffic conditions. Third, using the differences of flows and average velocities between the original simulation run and the altered capacity runs, we can evaluate the range and magnitude of the effects of small local infrastructural changes on a global scale. Such changes effect locations in the vicinity of the road segment, however, it will be interesting to measure the changes that occur in distant locations as they are typically considered negligible.

3.1 Super-sensitivity

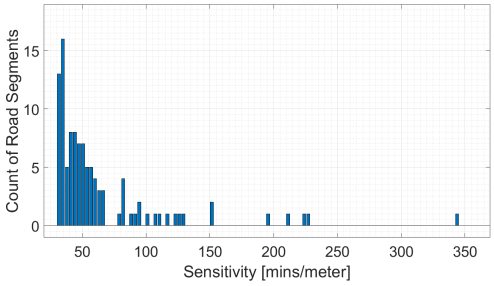
In order to measure the sensitivity of the system to capacity changes of a single road segment, we have defined a normalized value, which we call the sensitivity measure s_i of a road segment i with length l_i:

$$s_i = \frac{\left|T - T_i^{-1}\right| + \left|T - T_i^{+1}\right|}{2l_i} \tag{2}$$

The value represents the magnitude of change of travel time of the system, which is normalized by the length of the road segment, so the units of the measure are time difference per meter. Fig. 1 shows a histogram representing the distribution of sensitivity measures within the tested road segments. The distribution fits best a log-normal distribution, however, road segments can be observed to inhabit the far right portion of the distribution. Those segments can be identified as *super-sensitive*.



(a)



(b)

Figure 1: Histogram depicting the sensitivity measure of the tested road segments. Fig. 1a is the histogram of the complete generated dataset, while Fig. 1b is the histogram of road segments with sensitivity value of over 30 minutes per meter in order to better observe the *super-sensitive* region.

The log-normal distribution fitted to the generated sensitivity data set produces a mean $\mu = 1.67$ and standard deviation $\sigma = 0.8281$. The most sensitive location is outside the 5σ event zone, which is a strong indication that the distribution of sensitivity has a fat tail, where the road segments, which can be used as steering tools reside. One might think that roads with high throughput have a higher sensitivity value since changes in them affect more drivers. Our results show that this is not true as the correlation coefficient between the sensitivity measure and the throughput of the roads is 0.17. This means that there is almost no correlation between the number of vehicles that use a

road and how sensitive the system is to changes applied to it.

3.2 Braess Paradox

As described in the introduction section the Braess paradox occurs when capacity is taken away from a road segment and the traffic improves or alternatively when new capacity is created and congestion increases as a result. The paradox has been widely discussed in literature since it was first hypothesized, however, it has never been confirmed for a complete realistically sized system. The results that we have acquired can state with certainty that the paradox exists for the examined system, which aims to represent qualitatively correct traffic conditions in the city of Singapore. Fig. 2 represents the distribution of saved time for the tested links as a result of their capacity alteration.

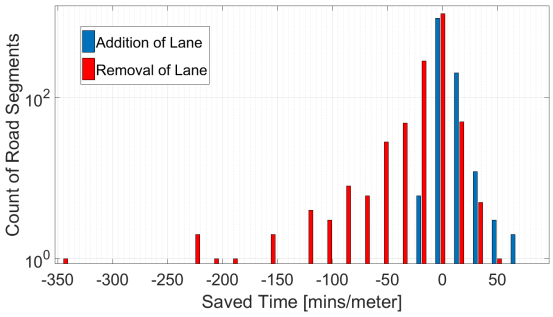
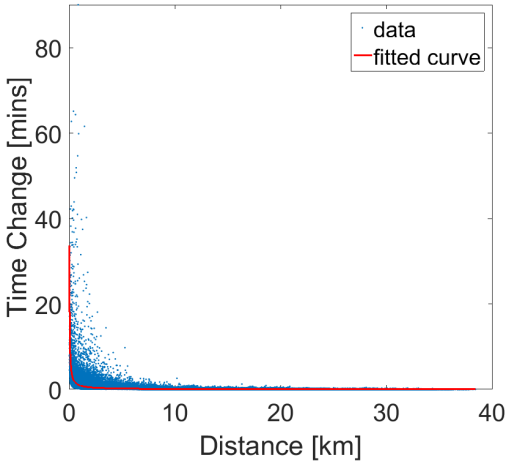


Figure 2: Distribution of saved time for the set of tested road segments in the case of both lane addition and lane removal.

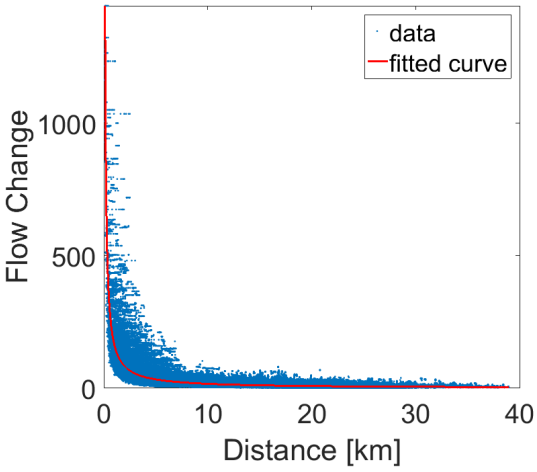
It can be observed that both the lane addition (blue) and lane removal (red) histograms cross the zero saved time point. In the far right portion of the lane removal histogram we can see that there is one road segment with a time save value of 50 minutes per meter. This means that if a lane is removed from this segment, the system travel time will be reduced by 50 minutes for every meter of removed lane. The same can be observed for the lane addition histogram. This means that when a lane is added to those roads the overall congestion of the network increases, although capacity has been added. There are in total 104 Braess roads, which should not receive new lanes, and 131 Braess roads, which should have their number of lanes reduced.

3.3 Range of effects

Local optimizations are often used both in transportation research literature and in practice. Such optimizations include introduction of new lane, traffic light phase optimization or pedestrian crossings. We hypothesize that, being complex systems, transportation systems might not react as predicted to local optimizations and secondary effects might be observed outside the scope of the examined area. We used the results of our simulations to visualize the change of travel time and flows versus the distance from the location that has been altered. Fig. 3 shows the resulting relationship. For every separate road configuration, we have compared 1) the total travel time along every road segment and 2) the flow on every segment, to the ones acquired from the original simulation. Every point on the graphs represents a comparison between a flow or travel time in the original simulation on a certain road segment and its counterpart from one of the alternative scenarios. After removing all points, which show no or minimal change, we have computed the function, which fits the data points and plotted it as well.



(a)



(b)

Figure 3: Magnitude of effect (3a change of average travel time and 3b change of flow) as a function of distance from point of road network alteration.

It can be observed that the relationship between the

magnitude of the effect and the distance is reciprocal. There, however, are some outliers that can be easily seen (points on the right side of the fitted line), which clearly confirm our hypothesis. For example, there is a road segment 6 km away from the alteration point, which exhibits a change with magnitude of 10 minutes of its overall travel time. Furthermore, there is a segment 3 km away from a point of alteration, which exhibits a change of throughput of $1,000$ vehicles. Changes of significant magnitude are observed in places up to 30 km from the alteration point. These findings indicate strongly that transportation systems should not be optimized locally as their complexity and high degree of interconnectivity and interdependence make it impossible to assume localized effects of system perturbations.

4 Conclusions

In this paper we have simulated UE traffic assignment for small road network changes in the sense of addition and removal of lanes from a set of road segments. The results obtained have been used in order to draw three main conclusions:

- Small infrastructural changes at certain road network locations have a disproportionally large effect on traffic conditions of the transportation system. Those locations have been labelled *supersensitive* and have a sensitivity value several standard deviations higher from the mean of the sensitivity value distribution

- There exist Braess road segments, which either worsen traffic conditions when their capacity is increased or improve it when the capacity is decreased, in a real world and real sized network, exhibiting realistic traffic demand.

- Small changes in the capacity of a road segment have significant effects not just on the region around the road segment but also non-negligible effects on distant locations of the system. Therefore, transportation systems should not be optimized locally.

As a future work, we would like to extend the formalism of the derivative or traffic assignment with respect to the road network as this can prove to be useful for designing optimal urban networks. Furthermore, we will continue our work on the concept of *supersensitivity* of roads and develop working strategies for their utilization. Last but not least, we would like to present even more solid proof of the distant effects of localized perturbations in transportation systems in order to adequately inform the community and the respective agencies, thus allowing them to implement only globally efficient optimization strategies.

Acknowledgments

This work was financially supported by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme and further computationally enabled by the resources provided by the HPI Future SOC Lab in Potsdam, Germany.

References

[1] L. T. Authority. Annual vehicle statistics 2014, 2014.

[2] M. Aymerich and A. Novo. Madrid critical intersections antiblocking strategies. In *Road Traffic Monitoring, 1992 (IEE Conf. Pub. 355)*, pages 70–, 1992.

[3] M. Beckmann, C. McGuire, and C. B. Winsten. Studies in the economics of transportation. Technical report, 1956.

[4] P.-D. D. D. Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12(1):258–268, 1968.

[5] S. C. Dafermos and F. T. Sparrow. The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards, Series B*, 73(2):91–118, 1969.

[6] M. Dongfang, S. Xianmin, T. Pengfei, and W. Dianhai. Critical traffic volume warrant of signal installing at equal weight intersections. In *Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on*, volume 1, pages 388–391, March 2011.

[7] R. L. Gordon. A technique for control of traffic at critical intersections. *Transportation Science*, 3(4):279–288, 1969.

[8] J. Ivanchev, H. Aydt, and A. Knoll. On identifying dynamic intersections in large cities. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2907–2914. IEEE, 2015.

[9] J. Ivanchev, H. Aydt, and A. Knoll. Spatial and temporal analysis of mismatch between planned road infrastructure and traffic demand in large cities. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1463–1470. IEEE, 2015.

[10] J. Ivanchev, H. Aydt, and A. Knoll. Information maximizing optimal sensor placement robust against variations of traffic demand based on importance of nodes. *IEEE Transactions on Intelligent Transportation Systems*, 17(3):714–725, 2016.

[11] J. Ivanchev, S. Litescu, D. Zehe, M. Lees, H. Aydt, and A. Knoll. Determining the most harmful roads in search for system optimal routing. Technical Report TUM-I1632, Technical University Munich, 2016.

[12] T. Roughgarden. *Selfish routing and the price of anarchy*, volume 174. MIT press Cambridge, 2005.

[13] Y. Sheffi. Urban transportation network. *Pretince Hall*, 1985.

[14] M. van Essen, T. Thomas, E. van Berkum, and C. Chorus. From user equilibrium to system optimum: a literature review on the role of travel information, bounded rationality and non-selfish behaviour at the network and individual levels. *Transport Reviews*, pages 1–22, 2016.

In-Memory Natural Language Processing: Fall/2016 Report of the HPI Future SOC Lab

Mariana Neves
Hasso Plattner Institute
August-Bebel-Str. 88
Potsdam, 14482, Germany
mariana.neves@hpi.de

Abstract

In this report we describe our activities during the last six months and that directly or indirectly profited from the SAP HANA instance provided by the HPI Future SOC Lab. Our activities in the NLP Lab in the EPIC chair of HPI covers a range of topics including implementation of NLP applications, development of domain resources and organization of events.

1 Introduction

The current data deluge demands fast and real-time processing of large datasets to support various applications, also for textual data, such as scientific publications, Web pages or messages in the social media. Natural language processing (NLP) is the field of automatically processing textual documents and includes a variety of linguistic and semantic-related tasks. Processing and semantically annotating large textual collection is a time-consuming and tiresome task which requires integration of various tools. In-memory database (IMDB) technology and the SAP HANA database comes as an alternative given its ability to quickly process large document collections in real-time and its built-in text analysis functionality. In this report we describe our activities during the last six months and that directly or indirectly profited from the SAP HANA instance provided by the HPI Future SOC Lab.

2 Activities

Our activities in the NLP Lab in the HPI/EPIC chair covers a range of topics including implementation of NLP applications, development of domain resources and organization of events. An overview of it is shown in Figure 1 and more information is provided in our Web page [1].

[1] https://hpi.de/de/plattner/projects/in-memory-natural-language-processing.html

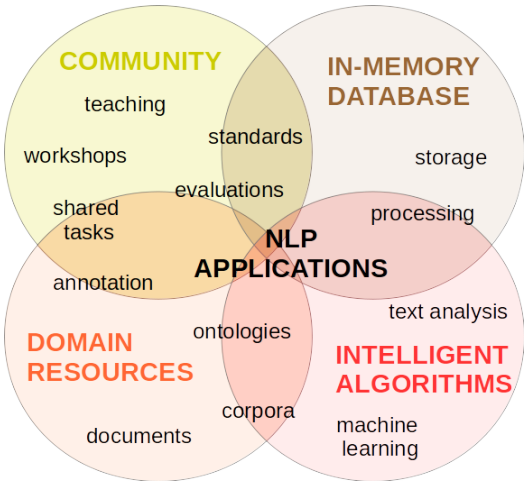


Figure 1: Overview of the activities are of the In-Memory Natural Language Processing Lab at HPI/EPIC chair.

Olelo NLP Platform. During the Bachelor project which took place in the 2015/2016 academic year, five students developed an intelligent system for browsing the scientific literature in biomedicine. The system was built on top of the HANA database and currently indexes more than 15 millions abstracts from the PubMed database of biomedical publications and more than 1.5 million full texts from PubMed Central Open Access subset. Moreover, we integrated two important resources in biomedicine: the MeSH ontology and the UMLS database, which includes various biomedical terminologies.

We used some of the built-in text analysis functionality in HANA, such as tokenization, part-of-speech tagging and dictionary-based named-entity recognition based on both MeSH and UMLS. We implemented further NLP algorithms in our platform, namely, question processing through natural language understanding, answer processing, ranking of documents according to the extracted concepts, and automatic text summarization. Figure 2 shows a screenshot of

the answers returned by Olelo for a particular question.

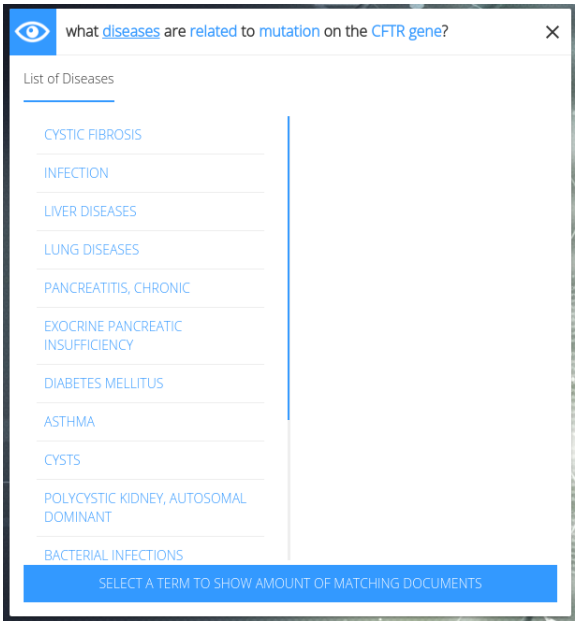


Figure 2: List of answers returned by Olelo for a question.

In particular, our summarization system was further evaluated on two uses cases: generation of ideal answers for questions in the BioASQ challenge and automatic summaries for genes [5]. Further, we have also published automatically generated summaries for genes in response for the hashtag #GeneOfTheWeek [2] promoted by the Ensembl database in Twitter. Each week, Ensembl names a gene and many research groups publish further information about the gene. One of such summaries is show in Figure 3.

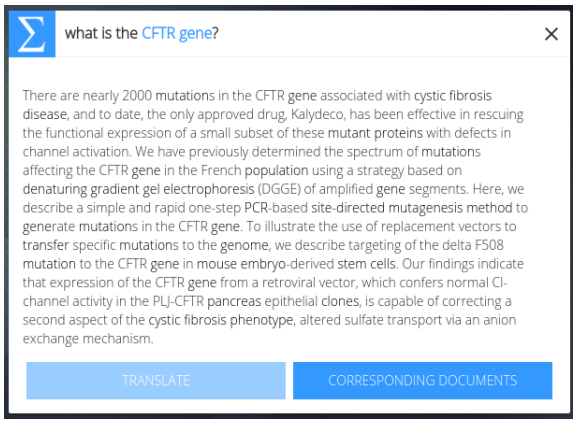


Figure 3: Summary for the CFTR gene.

TextAI annotation tool. In the context of a Master project, we developed an annotation tool which is powered with active learning [2]. TextAI learns from the manual annotation performed by the users and provides predictions (suggestions) for new documents. Current implementation of the named-entity recognition is based on the dictionary-based approached provided by HANA while the relation extraction uses machine learning algorithms available in the Predictive Analysis Library and the R library.

Resources. We also made use of the HANA database to support development of domain resources. Such resources are important to support development and evaluation of the Olelo platform. These are also valuable resources for te BioNLP community.

We created the first parallel collection of scientific publications for biomedicine [4]. The documents were retrieved from the Scielo database and we run several NLP tasks, such as language detection, sentence splitting and sentence alignment in order to create parallel corpora for three language pairs; English/Spanish, English/Portuguese and English/French. We also annotated more than 600 questions from the BioASQ dataset that will support training and evaluation of the question and answer processing components [3].

Shared Tasks. We participated for the third time in the BioASQ challenge with a question answering system previously developed in our chair as well as our new Olelo system [6]. We achieved very good results for the snippet retrieval and for the ideal answers as presented in the award (cf. Figure 4).

We organized a biomedical translation task in the First Conference of Machine Translation (WMT'16) based on the Scielo corpus [1]. We had participants from four countries in Europe and they contributed to solutions for all language pairs.

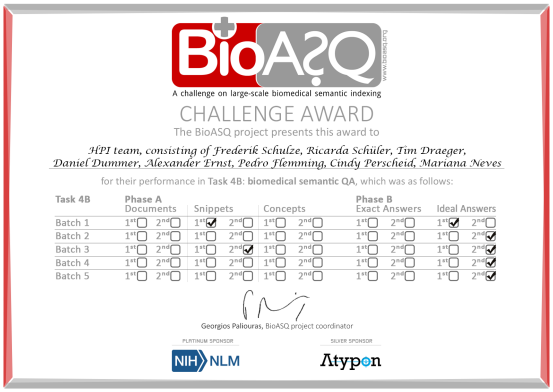


Figure 4: Award for our results in the BioASQ challenge 4.

[2] https://twitter.com/hashtag/GeneOfTheWeek?src=hash

3 Future Work

We have plans to improve the methods behind our Olelo question answering system. In particular, we aim to better understand the questions and better extract the answers through integration of semantic role labeling. Further, we are currently collecting more than 40 biomedical corpora to perform a comprehensive evaluation of our NER approach. These corpora will be later normalized regarding their semantic types and this information will be made available to the BioNLP community. Finally, we are also performing an evaluation of the system with external partners and implementing the functionality accordingly.

Regarding the TextAI tool, our future work focus on the implementation of a machine learning algorithm based on lazy learning approach. Such algorithms, such as example-based reasoning (EBR), do not perform any model training on real time but rather make prediction online based on previous training data. Finally, we also plan to evaluate TextAI regarding its application for corpus construction or data curation in collaboration with external partners.

Finally, we plan to organize again the biomedical translation task in WMT'2017. Therefore, we need to increase the size of the Scielo corpus and process the additional documents. We might also include biomedical documents from other sources, besides the Scielo database, and also additional language pairs.

References

[1] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Neveol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics.

[2] M. Grundke, J. Jasper, M. Perchyk, J. Sachse, R. Krestel, and M. Neves. Textai: Enhancing textae with intelligent annotation support. In *Proceedings of the Seventh International Symposium on Semantic Mining for Biomedicine (SMBM)*, pages 80–84, 2016.

[3] M. Neves and M. Kraus. Biomedlat corpus: Annotation of the lexical answer type for biomedical questions. In *Open Knowledge Base and Question Answering Workshop at the 26th International Conference on Computational Linguistics (Coling)*, 2016.

[4] M. Neves, A. J. Yepes, and A. Névéol. The Scielo Corpus: a Parallel Corpus of Scientific Publications for Biomedicine. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[5] F. Schulze and M. Neves. Entity-supported summarization of biomedical abstracts. In *Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining at the 26th International Conference on Computational Linguistics (Coling)*, 2016.

[6] F. Schulze, R. Schler, T. Draeger, D. Dummer, A. Ernst, P. Flemming, C. Perscheid, and M. Neves. Hpi question answering system in bioasq 2016. In *Proceedings of the Fourth BioASQ workshop at the Conference of the Association for Computational Linguistics*, pages 38–44, 2016.

Cloud-based Analytical Information Systems using RABIC

Oliver Norkus, Jürgen Sauer
Carl von Ossietzky Universität Oldenburg
Escherweg 2
26121 Oldenburg
{firstname.surname}@uni-oldenburg.de

Abstract

The combination of business intelligence (BI) and cloud computing (CC) is discussed increasingly in science and industry since years. Since the absence of standards and transparency encourages many organizations skepticism and incomprehension [2], [5]. The potential of standards in this area are already recognized [1]. This situation is now changed inasmuch as an integrated reference architecture for business intelligence in the cloud (RABIC) is available [6]. This reference architecure, we have tested and evaluated.

1 Introduction

Cloud-based Analytical Information Systems are a disruptive technology development. There are roots and precursor in both domains BI and CC. On the side of the cloud technology (such as grid computing, cluster computing, virtualization and outsourcing), and on the side of BI technology (such as core BI systems, mobile BI, adaptive BI). Nevertheless, BI in the cloud is a new technology bundle for providing personalized and configurable, scalable and flexible analytical IT services [5].
In various literature reviews, we found that all assessments and publications see great opportunities and potential for BI in the cloud (see for example [2], [1], [3], [4]). Some references formulats concerns, other focus on further challenges, for example in the field of security. But the common thread is that there BI in the Cloud is a future field which is still in its very early stages regarding the usage and standardization.
One of the reasons is that there are only few reports about experience. The absence of standards and transparency promotes skepticism and incomprehension. A lack of comparison and assessment models for cloud-based BI systems also fueled the uncertainty. First products of different manufacturers exist on the market, a non-comprehensive utilization is imperceptible as the tools are mostly heterogeneous, they cannot be combined in any way or integrate with existing systems. The main reasons are unanswered issues regarding the enterprise and software architecture. This emphasizes the potential of a standard.
This situation will now be completed: RABIC, the integrated reference architecture for business intelligence in the cloud, will improve this situation by supporting the standardization, increase the transparency and abolish the skepticism. The measures to receive these objectives are to make BI cloud services uniformly describable and comparable and to make BI cloud systems assessable, comparable, describable and uniformly implementable. The vehicles as the integral constituents of the reference architecture presented here are a taxonomy for BI cloud services and an architectural framework for BI cloud systems [6].
We have tested and evaluated this reference architecture in cooperation with the Hasso Plattner Institute HPI. In this report, we firstly (see section 2) explain what RABIC is and secondly (see section 3) we report from our experiments and evaluations. This reports ends with a short summary (see section 4).

2 RABIC

RABIC [6] aims to promote the standardization and to increase the transparency and understanding. To achieve this the systems and services in the BI Cloud environment are become uniformly and structured describable, comparable and assessable.
As a vehicle for this, the integrated reference architecture consisting of two artefacts [6]:

1. A consolidated taxonomy for BI Cloud services and
2. a standardized IT architecture for BI Cloud systems.

The taxonomy enables a uniform description and a structured comparison of BI Cloud services. Thus, providers can describe their BI Cloud product as well as customers can describe their demand uniformly. With the taxonomy as a comparison tool, the selection of a suitable service becomes easier. Thus, BI consultants and BI managers are supported by describing

their requirements and by finding the best matching service [4], [6].
The IT architecture serves as evaluation and comparison model for existing as well as design patterns for new BI Cloud systems. Thus, BI project managers, BI architects and BI developers are supported in their work, especially by comparison, selection, evaluation and development of BI Cloud systems. Figure 1 illustrates the relationship between the artifacts and the use of options.

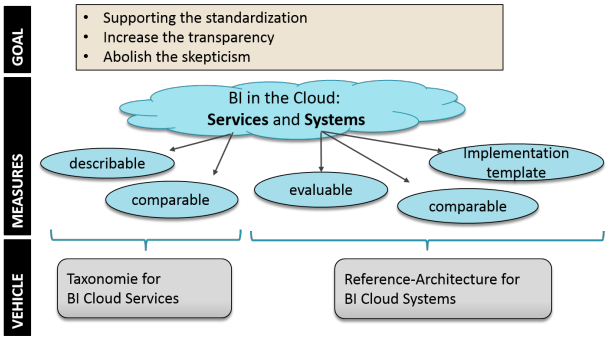


Figure 1: RABIC target range - in conformity with [6]

The classification of the integrated reference architecture is described following the approach from [**?**] where after a model is to be classified based on a *From which – Why – For what – For whom-* construct [6]:

- *From which?* The architecture is a model of BI systems and BI services that are based on the cloud technology.
- *Why?* The architecture serves to increase transparency and to promote standardization in order to favor acceptance and use of BI in the Cloud and to avoid the uncertainty and lack of experience.
- *For what?* The integrated reference architecture can be used for description, comparison and evaluation of existing BI cloud solutions. Besides supporting the IT architecture as an implementation template supports the operational development and use of BI in the Cloud. As a communication model it can be used to pass requests and expectations and enables discussions.
- *For whom?* Target user group of the reference architecture are BI architects and BI developers. These can use the model to compare existing product and to assess and set their own designs. Here, the IT architecture model supports especially to minimizing risks, to identify pitfalls and increases the efficiency of development and selection process.

The integrated nature of the reference architecture consists in its composition and the interaction of the

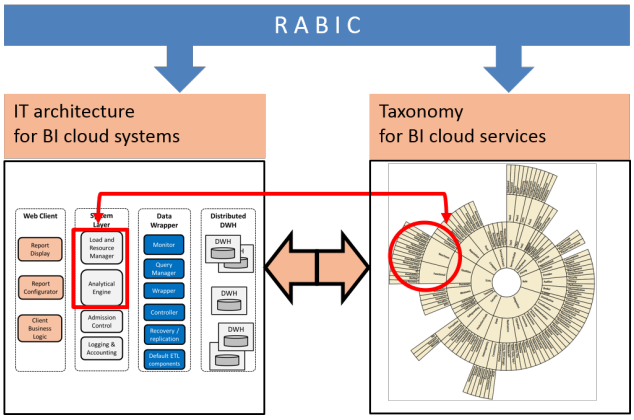


Figure 2: Integrated Reference Architecture for BI in the Cloud - in conformity with [6]

components. RABIC takes the service and the system level into account.

The *service layer* contains the taxonomy for describing and comparing BI Cloud services. Thus, providers can describe their service properties and buyers their service requirements, both uniformly and completely. This creates a better understanding of the services and a better result in the selection process. This taxonomy is a unique model (like a classicication schema) with that BI Cloud services can be sorted and categorized by certain criteria. The taxonomy has a hierarchical structure and is a formal model which can be validated [6].

The *system layer* embodies the system model with modular structure, component and class architecture patterns, data exchange formats and deployment diagrams. Therein varous scaling algorithms and billing models are discussed. In addition, there is a concept for a distributed data warehouse system. An abstraction mechanism allows any cloud and other storages as data warehouses. A rule-based algorithm selects the best storage service by optimizing the cost, performance and safety. This reusable system architecture makes recommendations in each area for implementing a BI Cloud system [6].

The interaction between the taxonomy and the architecture is an important characteristic of the integrated reference architecture: As Figure 2 illustrates, there is a clear and unambiguous link between the components and methods of the architecture and characteristics of services from the taxonomy. If the taxonomy is filled out for existing needs to a service, the corresponding components of the architecture can be used in order to satisfy the requirements. If there are existing services or existing systems, they can be described with the taxonomy and the architecture can be evaluated and compared with other approaches [6].

3 Evaluation

For testing and evaluation of RABIC, we have proceeded with case study within the realization of a prototype of RABIC. Furthermore we have performed expert interviews on RABIC in general and on the prototype in special.

By using ABIC, we implemented a BI cloud system. Thereby we used the methods, models and principles of RABIC. We have studied the individual components of the reference architecture and compared with our requirements. RABIC offers a modular structure, so that you can use components or emit them as you wish. The interactions and coordination between the remaining components in RABIC are well elaborated and worked.

Firstly we have analyzed our business needs and requirements regarding the BI Cloud service provided by the BI cloud system we want to implement. The taxonomy and the BI-Cloud Serivce Navigator (BI-CSN) gave us a good framework for explaning our needs. Secondly, directly derived from the needs, the features of our BI cloud service came out. By having a fullfilled BI-CSN, we could easily deduct - thirdly - the necessary architectural components for our implementation.

The prototype was developed by using state-of-the-art technologies provided by the Hasso-Plattner-Institute. For the local client we used the SAP UI 5 as frontend development tools. The virtual machines were running on a HP Converged Cloud. OpenStack has been used to implement the scalability guidelines of the reference architecture. As a database a SAP HANA Cloud Platform was used. The architectural design of the prototype is shown in Fig. 3.

Next to this implementation, we have used the RABIC specification as communication framework during our whole project. This gave us an uniform, understandable view in every project phase, especially for communicating with respective between different roles (e.g., project lead, product owner, developer, customer) . The running version of our prototype was applied in a case study wherein scenario-based architecture evaluations were performed. By this and by performing expert interviews, we have tested the BI cloud system.

The taxonomy as a description framework for BI cloud services can also be used for comparison of different BI cloud services, for example to help to deal with the varieties of BI cloud services within a selection process as a part of a launch. We also compared different services with each other. The taxonomy discovered all relevant aspects and differences and similarities were accessibly easily. Figure shows left and right fullfilled BI-CSN for BI cloud services and in the middle a visual comparison.

4 Summary

To help overcoming the prevalent skepticism of enterprise regarding BI in the Cloud, to promote the standardization and to increase the transparency, it is now reference architecture for BI in the cloud. This integrated reference architecture can be used to implement new BI cloud systems, to describe and evaluate existing BI cloud systems as well as to describe, to check and to compare BI cloud services [6].
By implementing a prototype based on the implementation template of RABIC, we have tested and evaluated the components of RABIC. On the system layer, the IT architecture is very well suited to be the base of a implementation of a new BI cloud system. The modular structure of the system architecture allows a adequately system complexity exact depending on the project requirements. During the comparison of our reseach prototype with an other prototype and with BI Cloud tools available at the market, we found that the RABIC system layer models are very suitable for comparison and evaluation of existing instances. The modular structure of the system architecture allows a system complexity exact depending on the requirements.
In overall, with RABIC now an integrated reference architecture exists which can be used as a description and comparison model, as an implementation template as well as an evaluation.

References

[1] O. Norkus. An approach for the standardization of business intelligence in the cloud. In U. Aßmann, B. Demuth, T. Spitta, G. Püschel, and R. Kaiser, editors, *Proceedings of Software Engineering & Management*, number 239 in LNI, pages 299–303. Bonner Köllen Verlag, 03 2015.

[2] O. Norkus and H.-J. Appelrath. Towards a business intelligence cloud. In *Proceedings of the Third International Conference on Informatics Engineering and Information Science (ICIEIS2014)*, pages 55–66. SDIWC, 09 2014.

[3] O. Norkus, B. Clark, F. Merkel, B. Friedrich, J. Sauer, and H.-J. Appelrath. An approach for a cloud-based contribution margin dashboard in the field of electricity trading. In D. Cunningham, P. Hofstedt, K. Meer, and I. Schmitt, editors, *Informatik 2015*. Bonner Köllen Verlag.

[4] O. Norkus and J. Sauer. A taxonomy for describing bi cloud services. In *Proceedings of the International Conference on Semantic Web Business and Innovation*, pages 1–12. SDIWC, 2015.

[5] O. Norkus and J. Sauer. Towards an architecture of bi in the cloud. In G. Silaghi, J. Altmann, and O. Rana, editors, *Economie of Grids, Clouds, Systems and Services (GECON2015)*. Springer, 09 2015.

[6] O. Norkus and J. Sauer. Rabic: A reference architecture for business intelligence in the cloud. *Journal of Communication and Computer*, (13):244–260, 2016.

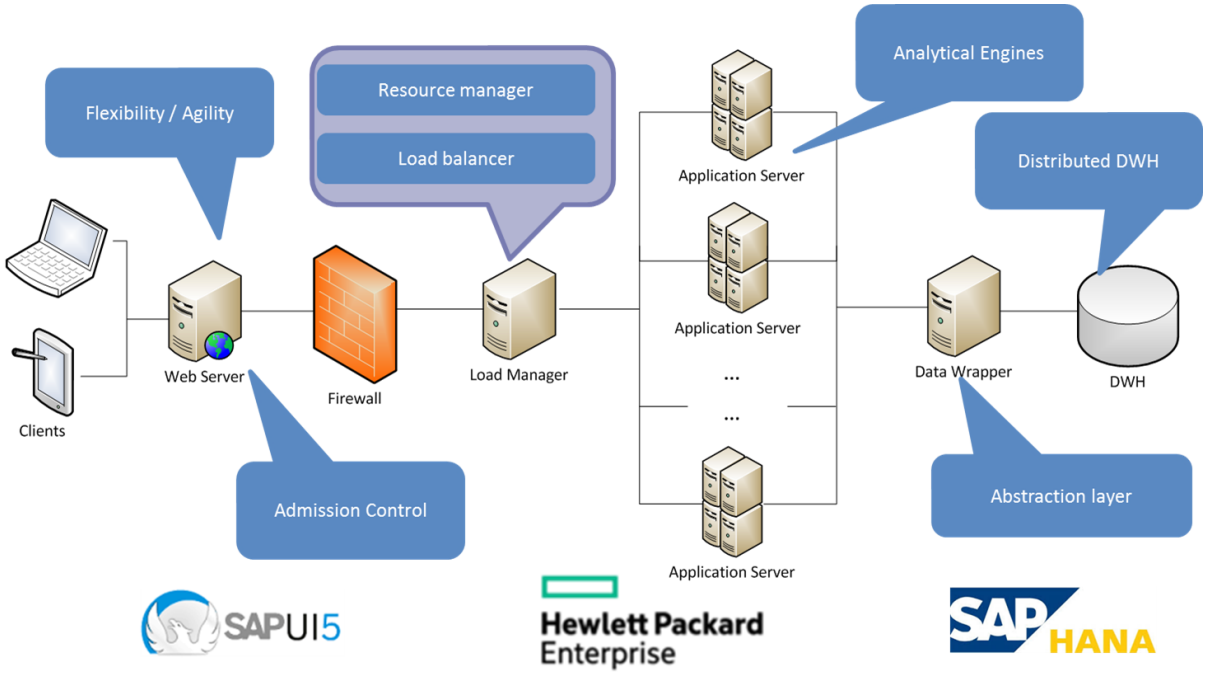


Figure 3: Architecture of the prototype - in conformity with [6]

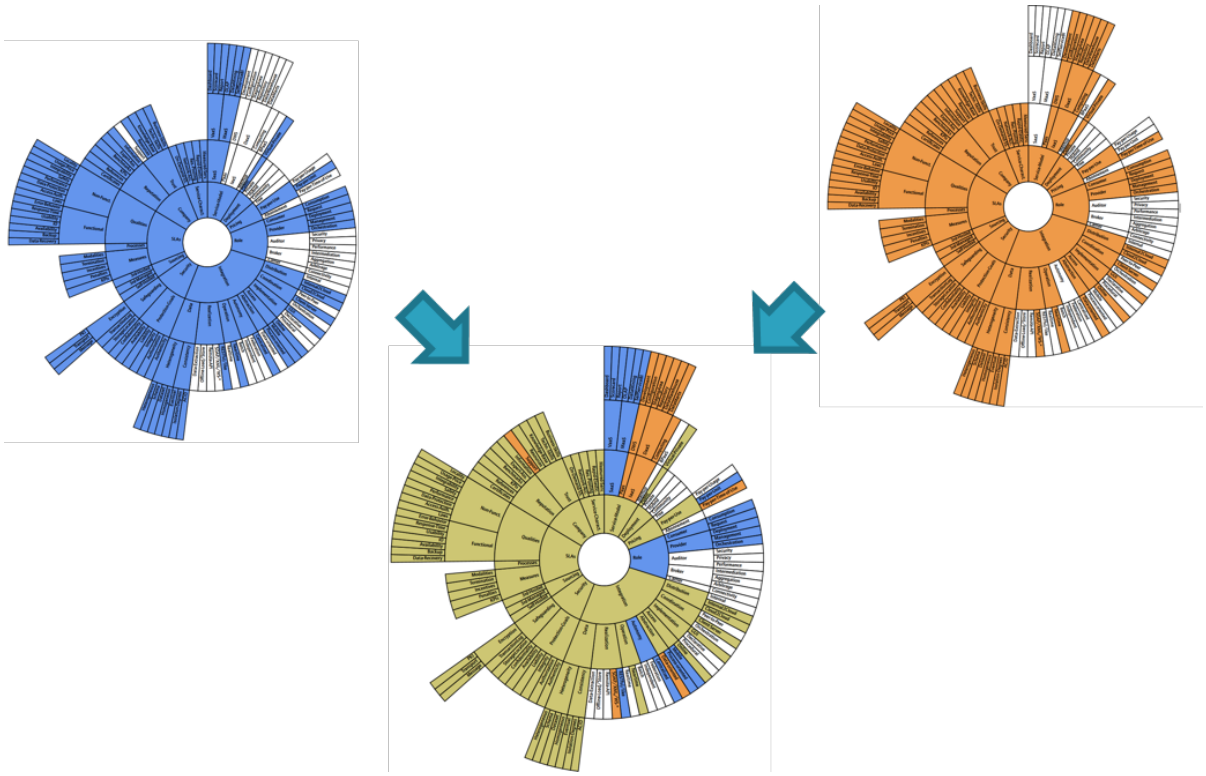


Figure 4: Visual comparion with BI-CSN - in conformity with [6]

High Performance Event Streaming and Security Analytics

Andrey Sapegin, David Jaeger, Feng Cheng, Christoph Meinel
Hasso Plattner Institute
University of Potsdam
PO Box 900460
14440 Potsdam, Germany
{feng.cheng, david.jaeger, martin.ussath, andrey.sapegin, marian.gawron}
@hpi.uni-potsdam.de

Abstract

The number of log events produced in large IT infrastructures grows up to multiple billions per day. In this report, we describe our research and development of a high performance event streaming mechanism that is able to deeply and rapidly normalize and persist logs (in HANA) by using distributed, parallel processing, and new inter-thread messaging technologies. This research was performed under FutureSOC Lab project. Utilising high-speed data processing and normalisation approach, we were able to process hundreds of million of security related data records from our partner (large enterprise company). Deep normalisation of this records allowed us to successfully apply our existing outlier detection methods and to detect previously unknown issues in the enterprise infrastructure.

1 Introduction

Large enterprise networks produce vast amounts of security related log messages, and their volume is constantly growing. Both SIEM and IDS should be able to handle such volumes to provide efficient protection for enterprise networks. According to Gartner [4], large SIEM setups process about 25,000 events per second. In practice, this number is often even higher [7, 1]. To deal with such amounts of messages, special techniques need to be developed.

In this report we first describe our high-speed normalisation approach, which is able to process extremely high volumes of log messages (up to 300,000 per second). To prove our concepts we apply newly developed techniques on a dataset from large enterprise company containing several hundreds million events.

Besides normalisation mechanism, under current Future SOC Lab project we have also developed a high-speed log analytics methods based on machine learning. After normalisation of log messages from selected dataset, we apply these machine learning methods to identify previously undetected security and configuration issues.

1.1 HPI Future SOC Lab resources

Processing of high amounts of log messages requires both hardware and software resources. To make this project accomplished, HPI Future SOC Lab provided us an access to several shared SAP HANA in-memory database instances, as well as an exclusive access to VMware ESXi hypervisor with 256 Gb RAM and 64 CPU cores. This resources were used during the project to estimate performance of multinode normalisation and to apply our anomaly detection algorithms on data to discover suspicious events, that cannot be identified using other methods.

1.2 Dataset

The dataset provided us for analysis from a large enterprise company contains 620,280,945 log messages that we produced within a period of 2 weeks. These messages came from various systems from enterprise infrastructure, including firewalls, IDSs, teleconference systems, etc. All these messages were stored in the same table, however, with only 2 fields filled in ('timestamp' and 'message'), where the last one contained all relevant information about each event, including IP addresses, hostnames, usernames and so on. Within 620 million events, only Cisco Nexus logs (approx. 74 million) were normalised and various fields such as hostname, ip_proto, net_dst_ip4, net_dst_port, net_src_ip4, net_src_port, tag_action, tag_subject were available for instant analysis.

2 Normalization

To be able to process messages from our dataset, the 'message' field should be parsed to extract all information necessary for the analytics. Our extraction mechanism is based on the hierarchical regular expressions

[2], that match messages of specific type and extract matched parts into pre-defined fields of our Object Log Format [6].

Under this project we created hierarchical regular expressions for 35 different log message patterns like Checkpoint, Tippingpoint, Cisco, F5, iptables, SSH, etc. Our normalisation module updates the table with original data and fills empty fields in. After this step, we were able to parse extra fields (depending on the pattern and log message), as presented in Figure 1

All these fields can be later used for the analysis, including our outlier detection based on machine learning [5].

However, since it was required to process several hundreds million events (that were collected just within time period of 2 weeks), we needed to apply hierarchical regular expression to our dataset with a very high speed. To achieve this goal, we have used 2 techniques for normalisation mechanism: a Disruptor pattern from LMAX [3] presented in Figure 2 and multi-node architecture.

Disruptor pattern, originally developed for high-speed transaction exchange on stock markets, allows high-speed communication between normalisation threads thanks to original lock-free exchange of messages between threads. The resulting performance is presented in Figure 3.

Figure 3 shows, that the use of a Disruptor pattern allows us to reach a normalisation performance of 145 thousand of messages per secon. After we have implemented multi-node normalisation, we were able to almost double this performance up to 264 thousand of messages per second, please see Figure 4 for details.

Thus, we were able to normalise 498,865,953 log messages from total of 620,280,945 log messages within a short time. More detailed description of event normalisation process can be found in our paper [3].

After we applied all available regular expressions (not only checkpoint ones) on the data, we are able to parse different types of log messages, that are presented in Table 1.

All in all, with regular expressions we have created for the selected dataset, we are able to correctly normalise almost all Cisco, f5, Checkpoint and Tippingpoint log messages.

3 Issues detected on normalised data

From the Table 1, we can conclude, that the dataset mainly contains firewall log messages. This firewall messages already contain some attack alerts and other issues. So, before we apply anomaly detection on the data, we first collect information about detected issues using database queries. Please see subsections below for details.

Type	Number of events
syslog	120781714
syslog-checkpoint_base	884237
syslog-checkpoint_base-checkpoint_message	184314360
syslog-cisco_access_list	21295229
syslog-cisco_deny_inbound_UDP	3002772
syslog-cisco_deny_tcp	5897301
syslog-cisco_deny_tcp_reverse_check	9467016
syslog-cisco_os_base	5306098
syslog-cisco_os_base-cisco_check_copy_to_logflash	5768481
syslog-cisco_os_base-cisco_flapping_ports	16618535
syslog-cisco_os_base-cisco_ios	397912
syslog-cisco_os_base-cisco_ios-cisco_nexus_list	49504433
syslog-cisco_os_base-cisco_nexus_access	53222476
syslog-cisco_os_base-cisco_nexus_list	22339887
syslog-cisco_teardown_dynamic_translation	90732
syslog-cisco_telepresence_vcs_1-cisco_telepresence_vcs_1_msg	12141701
syslog-cisco_telepresence_vcs_2	31898736
syslog-dns_request	4408743
syslog-f5_ssl_acc	5626886
syslog-f5_ssl_req	5617328
syslog-iptables-88	263
syslog-iptables-90	290
syslog-pam_authentication_failure	149
syslog-pam_session_closed	283081
syslog-pam_session_opened	281800
syslog-service_startup_succeeded	28
syslog-ssh_password_accepted	146
syslog-ssh_password_rejected	131
syslog-syslog	594
syslog-syslog_repeated2	5833019
syslog-tippingpoint_sms	8072010
syslog-tis_httpgw_connect	3146549
syslog-tis_httpgw_deny	4390090
syslog-tis_httpgw_disconnect	3135632
syslog-tis_httpgw_exit	11999379
syslog-tis_httpgw_permit	23919929

Table 1: Number of normalised log messages by type

- SEVERITY
- APPLICATION_PATH
- APPLICATION_PROTO
- COMMAND
- PID
- STATUS
- SRC_HOST
- NET_SRC_IP4
- NET_SRC_MAC
- NET_SRC_PORT
- NET_SRC_PIF
- NET_SRC_VIF
- DEST_HOST
- NET_DST_IP4
- NET_DST_MAC
- NET_DST_PORT
- NET_DST_PIF
- NET_DST_VIF
- IP_LEN
- IP_PROTO
- PRODUCT
- HOSTNAME
- TAG_ACTION
- TAG_SUBJECT
- SUBJECTUSER_USERID
- TARGETUSER_USERID
- FILE_PATH

Figure 1: Extra fields extracted from data using hierarchical regular expressions

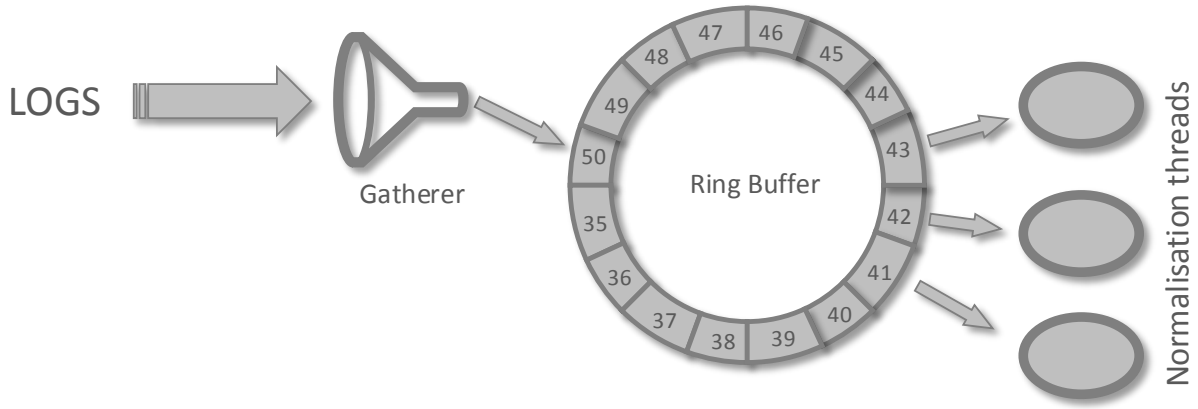


Figure 2: Ring Buffer used in the Disruptor lock-free pattern for communication between threads

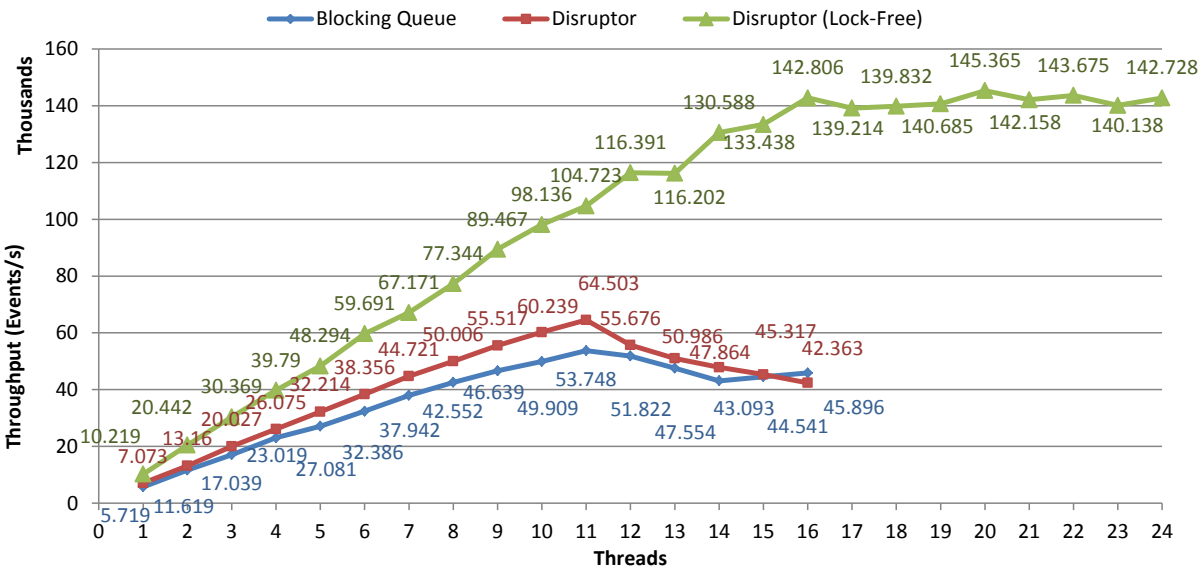


Figure 3: Perfomance of normalisation mechanism using Disruptor approach

3.1 Authentication failures

From the Table 1 it is possible to see that there are 149 PAM authentication failures. Thanks to normalisation, we are able to quickly filter out these messages by normalisation pattern ID. 125 of these failures are related to a single remote host trying to login on 2 different corporate servers over SSH and HTTP with different usernames (guest, admin, root, suse-gm, vmware), which looks like a part of brute-force attack. Other 14 messages are probably not harmful, since they present a few standalone authentication issues and probably relate to "normal" mistyped password events.

3.2 Attacks detected by Checkpoint and Tippingpoint

Besides password authentication failures, log messages also contain alerts from Checkpoint and Tippingpoint devices. In the Checkpoint firewall data (185,198,597 events normalised) we have identified 747825 alerts in total distributed between 10 types of

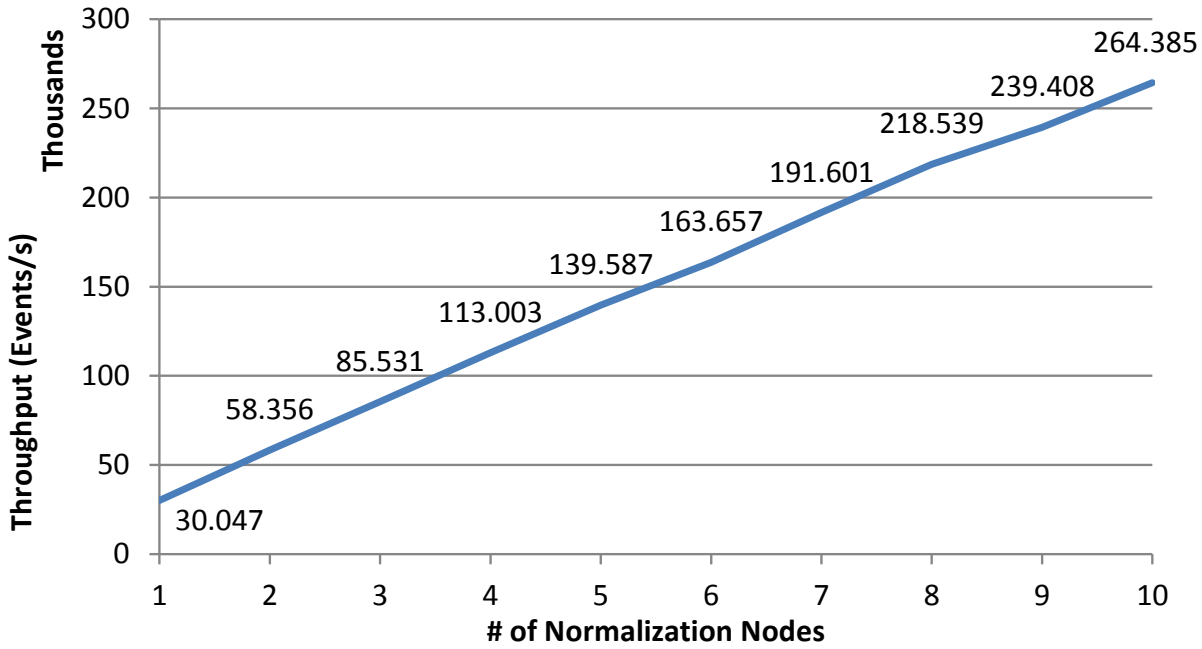


Figure 4: Perfomance of multi-node normalisation mechanism

alerts. 8,072,010 events from HP Tippingpoint Security Management System contained 7425649 alerts distributed within 394 alert types. Such high amount of alert types probably indicates that the enterprise network is regularly audited with vulnerability scanners such as Nessus. The detected alerts are therefore mainly not harmful and present audit traces.

3.3 Anomaly Detection results on Cisco Nexus access logs

After we have checked authentication failures and attacks detected by firewalls in the dataset, we have selected several event types to prove our outlier detection approach. We started our analysis with data subset containing 70 million of Cisco Nexus logs. To perform anomaly detection, we have applied a hybrid anomaly detection approach, based on spherical k-means and one-class SVM. As a algorithm's output, we have got clusters with suspicious events. The raw data showed that similar events were correctly clustered together. Each cluster is prepended with a decision value from SVM ensemble, which is a score for ranking. So the top-placed clusters in the output are most suspicious ones.

We have manually re-checked first 1000 most suspicious events and found out 1 suspicious type of event, namely blocked OSPF packets. The OSPF messages are usually exchanged only between OSPF neighbours, which should be pre-configured. Thus, blocked OSPF messages indicate either a firewall or router misconfiguration.

3.4 Anomaly detection on all *cisco_deny* messages

Next, we decided to check anomalous events in Cisco ASA logs (cisco_asa_deny). We have applied the same hybrid anomaly detection on 18,367,089 of Cisco ASA messages. The top-ranked anomaly cluster from the algorithm's output contains failed telnet connections. We have manually checked the full dataset and discovered, that it contains only 58 denied telnet connections and only to 2 unique IP addresses.
Since telnet protocol is almost never used nowadays, such small amount of denied telnet messages could indicate a personalised attack on the specific servers within enterprise network.
Thus, we have demonstrated, that usage of our outlier anomaly detection allows to identify suspicious log messages, which are hard to detect or filter within huge amounts of firewall logs otherwise.

4 Conclusion

Under this project we have implemented and proved on the real dataset several important concepts. First of all, we have developed a high-speed deep normalisation for security-related log messages, based on Disruptor pattern and multi-node architecture. To make application high-speed normalisation possible, we have created regular expression rules and normalised several log sources, such as *checkpoint*, *cisco*, *f5*, *tippingpoint*. This normalisation allows to use extracted fields such as IP address, port number, hostname, etc. as a features for anomaly detection algorithm and also allows to filter data by log type. Our outlier detection approach based on machine learning

techniques such as spherical k-means and SVM ensemble let us to detect several issues in the data that were previously undetectable with other methods as well as firewalls and IDSs.

References

[1] S. Bhatt, P. K. Manadhata, and L. Zomlot. The Operational Role of Security Information and Event Management Systems. *IEEE Security & Privacy*, 12(5):35–41, sep 2014.

[2] D. Jaeger, A. Azodi, F. Cheng, and C. Meinel. Normalizing Security Events with a Hierarchical Knowledge Base. volume 9311 of *Lecture Notes in Computer Science*, pages 237–248. Springer International Publishing, Cham, 2015.

[3] D. Jaeger, A. Sapegin, M. Ussath, F. Cheng, and C. Meinel. Parallel and distributed normalization of security events for instant attack analysis. In *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8. IEEE, dec 2015.

[4] O. R. Kelly M. Kavanagh. Magic Quadrant for Security Information and Event Management Market Definition / Description. Technical report, Gartner, 2015.

[5] A. Sapegin, M. Gawron, D. Jaeger, F. Cheng, and C. Meinel. Evaluation of in-memory storage engine for machine learning analysis of security events. *Concurrency Computation*, pages n/a–n/a, 2016.

[6] A. Sapegin, D. Jaeger, A. Azodi, M. Gawron, F. Cheng, and C. Meinel. Hierarchical object log format for normalisation of security events. In *2013 9th International Conference on Information Assurance and Security (IAS)*, IAS '13, pages 25–30. IEEE, dec 2013.

[7] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda. Beehive. In *Proceedings of the 29th Annual Computer Security Applications Conference on - ACSAC '13*, ACSAC '13, pages 199–208, New York, New York, USA, 2013. ACM Press.

Research of Ensemble Learning Techniques for SAP HANA and Development of a Benchmark System

Sabrina Plöger
FH Dortmund
Emil-Figge-Str. 42
44227 Dortmund, Germany
sabrina.ploeger@fh-dortmund.de

David Müller
FH Dortmund
Emil-Figge-Str. 42
44227 Dortmund, Germany
david.mueller@fh-dortmund.de

Christoph M. Friedrich
FH Dortmund
Emil-Figge-Str. 42
44227 Dortmund, Germany
christoph.friedrich@fh-dortmund.de

Christoph Engels
FH Dortmund
Emil-Figge-Str. 42
44227 Dortmund, Germany
christoph.engels@fh-dortmund.de

Abstract

Ensemble methods (like random forests, quantile forests, gradient boosting machines and variants) have demonstrated their outstanding behavior in the domain of data mining techniques.

This project focuses on literature research and development of ensemble learning methods, in order to propose strong techniques to be considered for further extensions of the SAP HANA PAL library.

Furthermore, a benchmark system is developed, to compare such data mining methods from PAL with algorithms from the R environment.

1 Project Idea

In the first five Future SOC Lab periods, the University of Applied Sciences and Arts Dortmund successfully addressed the topic *Data Mining on SAP HANA* with their projects *Raising the power of Ensemble Techniques* and *Performance Optimization of Data Mining Ensemble Algorithms on SAP HANA* [1][2]. The initial project idea was to compare different opportunities, which enable the usage of predictive analytical techniques on SAP HANA.

SAP is offering the Predictive Analysis Library (PAL), which contains more than 70 well-known algorithms in the fields of classification analysis, association analysis, data preparation, outlier detection, cluster analysis, time series analysis, link prediction and others [3].

Starting with basic comparisons between SAP HANA PAL algorithms and R algorithms, they proceeded with implementing own data mining algorithms in different languages on SAP HANA. The final programming result is a random forest implementation in C++, which was introduced in spring 2014. This kind of ensemble learning algorithm was not available in the PAL library in these days and thus it gave greater opportunities for analyzing data stored on SAP HANA. In comprehensive tests, the random forest implementation evinced itself as a strong and fast algorithm with convincing prediction results. [4]

The project idea of the recent and upcoming Future SOC Lab periods is to support the SAP PAL development team from Shanghai directly. In a first step, results of the preceding projects are utilized by delivering the random forest with additional documentations of the underlying concepts. In a second step, comprehensive research analysis of ensemble learning techniques is performed in order to support the PAL development team in the process of selecting and implementing ensemble methods for the PAL library. The third and last step comprises the conception and implementation of a benchmark system, to make PAL methods comparable with methods from the R environment and other libraries.

Why Ensemble Methods?

Predictive statistical data mining has evolved further over the recent years and remains a steady field of active research. The latest research results provide new data mining methods, which lead to better results in model identification and behave more robustly especially in the domain of predictive analytics. Most

analytic business applications lead to improved financial outcomes directly, for instance demand prediction, fraud detection and churn prediction [5][6][7][8][9][10]. Even small improvements in prediction quality lead to enhanced financial effects. Therefore, the application of new sophisticated predictive data mining techniques enables business processes to leverage hidden potentials and should be considered seriously.

Especially for classification tasks ensemble methods (like random forests) show powerful behavior [11][12][13] which includes that

- they exhibit an excellent accuracy,
- they scale up and are parallel by design,
- they are able to handle
    - thousands of variables,
    - many valued categories,
    - extensive missing values,
    - badly unbalanced datasets,
- they give an internal unbiased estimate of test set error as primitives are added to ensemble,
- they are robust to overfitting,
- they provide a variable importance and
- they enable an easy approach for outlier detection.

What are Ensemble Methods?

The main idea of ensemble methods is to combine a set of models, in order to obtain a better composite global model, which can reach more accurate and reliable estimates or decisions than one single model. This base learning algorithm can be a decision tree or any other learning algorithm.

Generally, it can be distinguished between two kinds of ensemble methods. *Independent ensembles* on the one hand, consist of models which can be trained independently from each other. Thus, the construction can be easily parallelized in appropriate multi-core environments like SAP HANA. Examples for the independent ensemble methodology are bagging and random subspace ensembles. [14] On the other hand, the models of a *dependent ensemble* mutually influence each other and are therefore interdependent. Because each model uses the knowledge of the previous models to adjust its construction, the ensemble has to be trained sequentially. Famous representatives of dependent ensemble methods are for example adaBoost, stochastic gradient boosting and iterated bagging. [15]

2 Used Future SOC Lab Resources

For this project a SAP HANA instance with the latest PAL revision is provided on a virtual machine as well as an R server on a same sized virtual environment.

3 Findings and Deliveries

The main deliveries of this project are the handover of the random forest implementation and corresponding documentation, the results of the literature review on ensemble learning techniques as well as the developed benchmark system.

In the mentioned literature analysis, a selection of innovative and state-of-the-art ensemble algorithms are investigated and evaluated. Each ensemble method is considered with regard to its process, availability, application and performance.

The benchmark system on the other hand supports the PAL development team by giving the opportunity to compare new developed and integrated methods from the PAL library with methods from the R environment or other libraries. The benchmark system is able to execute any PAL or R method with different parameter settings and monitors the model performance as well as the system utilization during runtime.

4 Next Steps

The main objective of this project is to support the PAL team in perspective research and development tasks. The cooperation between the project team of the University of Applied Sciences and Arts Dortmund and the SAP HANA PAL team was terminated for one year and ended in September 2016. In the following, potential activities are listed, built on the results of this project.

Activity "Implementation of selected ensemble methods"

In the recent project period comprehensive research was carried out to determine strong ensemble prediction models and to select certain methods for implementation. In the next step, these models must be developed and integrated into the PAL library.

Activity "Extension of the SAP Benchmark System"

The handover of the developed benchmark system was the third and last milestone of the cooperation project with SAP. There are different opportunities to extend the existing implementation:

- Enrich the system with more datasets to be able to execute more complex test scenarios.

- Enrich the system with a wider range of methods by developing appropriate method interfaces.
- Enrich the system with further connections to other environments. At the moment, PAL and R methods can be executed, but the benchmark system can be extended to connect to any other data mining library as well.

Activity "Open research tasks"

There are still open research topics which were not considered in the preceding project periods. Thus, research on the following topics might be carried out:

Anomaly detection

Anomaly detection, also known as outlier detection, is a subarea of data mining. The goal is to identify untypical or conspicuous data in a dataset. In practice, users are facing different problems by applying those methods, as for example choosing the right method and adjusting its parameters or dealing with sparse and bad labeled datasets. This activity comprises the identification of advantages and disadvantages of those methods, the determination of strong algorithms and their implementation on SAP HANA. [16]

Machine learning on sparse data

Many data mining methods do not work well on sparse data [17]. The goal is to carry out research on state-of-the-art solutions and to implement a selection of those algorithms on SAP HANA.

5 References

[1] C. Engels, C. Friedrich: "Proposal - Raising the power of Ensemble Techniques", Proposal to summer 2013 period at the HPI Future SOC Lab, 2013.

[2] D. Müller, C. Engels, C. Friedrich: "Proposal - Performance Optimization of Data Mining Ensemble Algorithms on SAP HANA", Proposal to summer 2014 period at the HPI Future SOC Lab, 2014.

[3] SAP AG: "SAP HANA Predictive Analysis Library (PAL) (document version: 1.0 – 2015-11-25)", 2015, URL: http://help.sap.com/hana/SAP_HANA_Predictive_Analysis_Library_PAL_en.pdf, accessed on 22.03.2016.

[4] D. Müller, S. Plöger, C. Engels, C. Friedrich, "Optimization of Data Mining Ensemble Algorithms on SAP HANA", Project report to summer 2015 period at the HPI Future SOC Lab, 2015.

[5] R. E. Banfield et al.: "A Comparison of Decision Tree Ensemble Creation Techniques", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 1, 2007.

[6] S. Benkner et al.: „@neurIST Infrastructure for Advanced Disease Management through Integration of Heterogeneous Data, Computing, and Complex Processing Services", DOI:10.1109/TITB.2010.2049268, IEEE Transactions on Information Technology in BioMedicine, 14(6), Pages 1365 - 1377, 2010.

[7] C. Engels: "Basiswissen Business Intelligence", W3L Verlag, Witten, 2009.

[8] C. Engels; W. Konen: "Adaptive Hierarchical Forecasting".Proceedings of the IEEE-IDACCS 2007 Conference, Dortmund, 2007.

[9] J. Friedman: "Computational Statistics & Data Analysis", Volume 38, Issue 4, 28 February 2002, Pages 367–378, 2002, URL: http://dx.doi.org/10.1016/S0167-9473(01)00065-2, accessed on 22.03.2016.

[10] G. Üstünkar et al.: "Selection of Representative SNP Sets for Genome-Wide Association Studies: A Metaheuristic Approach", DOI:10.1007/s11590-011-0419-7, Optimization Letters, Volume 6(6), Pages 1207-1218, 2012.

[11] L. Breiman: "RF / tools – A Class of Two-eyed Algorithms", SIAM Workshop, 2003, URL: http://www.stat.berkeley.edu/~breiman/siamtalk2003.pdf , accessed on 22.03.2016.

[12] L. Breiman: "Random Forests", 1999, URL: http://www.stat.berkeley.edu/~breiman/randomforests-rev.pdf, accessed on 22.03.2016.

[13] G. Seni, J. Elder: "Ensemble Methods in Data Mining", Morgan & Claypool, San Rafael, California, 2010.

[14] Z. Zhou.: "Ensemble Methods. Foundations and Algorithms" CRC Press, Hoboken, 2012.

[15] L. Rokach, O. Maimon.: "Data mining with decision trees. Theory and applications." World Scientific, Singapore, 2008.

[16] C. Aggarwal: "Outlier Analysis", Springer, New York, 2013.

[17] T. Hastie, R. Tibshirani, M. Wainwright: "Statistical Learning with Sparsity: The Lasso and Generalizations", Chapman & Hall/CRC Monographs on Statistics & Applied Probability, CRC Press, Florida, 2015.

Follow-Up Project: Sentiment Analysis on Twitter Data Using Entity and Fact Extraction

Marlene Knigge
Technical University of Munich
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
marlene.knigge@in.tum.de

Christopher Kohl
Technical University of Munich
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
christopher.kohl@in.tum.de

Galina Baader
Technical University of Munich
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
galina.baader@in.tum.de

Harald Kienegger
Technical University of Munich
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
harald.kienegger@in.tum.de

Helmut Krcmar
Technical University of Munich
Chair for Information Systems
Boltzmannstr. 3, 85748 Garching, Germany
krcmar@in.tum.de

Abstract

The goal of our project is to extend the outcomes of our former project[1] in which we identified perceived risks and emotions of autonomous driving from Twitter data ("Tweets"). On this social media platform, the topic of autonomous driving is discussed brightly. Positive or negative emotions may be justified in some cases, in others we cannot recognize any rational reason. In the first project, we extracted sentiments from tweets dealing with autonomous driving with native SAP HANA, PAL, and R text mining algorithms. Now we applied entity and facts extraction provided by SAP HANA Text Analysis, improved our data pre-processing, and customized the configurations for generating the Text Analysis Fulltext Index. Moreover, we built a web interface for visualizing the results which offers near real-time access to new tweets. Lastly, we describe the limitations of our work.

1 Introduction

Autonomous driving is a brightly discussed topic, especially, in case of accidents with autonomous driving cars involved, e.g., the Google Car or Tesla [2, 3]. These are not only to be found in the news, but also discussed in social media like Twitter. Today, people from all over the world can access this platform and spread their thoughts, opinions, and concerns by posting tweets – text messages with a maximum of 140 characters which can be read by anyone with a Twitter account. By doing so, each day (velocity) they generate huge amounts (volume) of unstructured (variety) data: big data [4]. Companies, e.g. the automotive industry, may generate business value from this freely accessible data by extracting information relevant to their business or they may use social media to transmit news and information to (potential) customers. Listening to this Voice of Customer (VoC) is much faster and can be more efficient than using outdated methods like asking customers for it face to face or via telephone surveys [5]. Hardware, applications, and algorithms have improved over time. In-memory databases and new frontend tools allow to analyze these huge amounts of data and to extract valuable information from it.

Nevertheless, for an effective analysis of big data, it is important to use not only computer power but to combine it with human minds. Therefore, it is crucial to present data in a way which can be captured and understood easily by human beings. As the human vision can gather more information than all other human senses combined, graphical representations of data

[1] For further information, please read the project report of our former project at HPI Future SOC Lab (Spring 2016) [1].

provide the largest stream of information between a human and a computer [6]. Moreover, visualizations can be understood and interpreted very effectively by humans [7].

In this project, we want to improve our analysis of social media data concerning the emerging technology of autonomous cars.

2 Project

In the first project[2] we started conducting sentiment analyses on Twitter data concerning autonomous driving. We will describe our outcomes in the next section before we elaborate the goal and design of this current project.

2.1 Previous Project Results

In our former project we executed sentiment analysis on Twitter data using Text Analysis, Text Mining, and different Predictive Analysis Library (PAL) algorithms. We got the best results using the Voice of Customer extraction provided by Text Analysis, and the K-Nearest Neighbors algorithm (KNN) of Text Mining.

Using Text Analysis, it was possible to analyze single parts of tweets and identify sentiments and problems. Applying the algorithm to a training data set containing about 7,500 rows about autonomous cars showed that people perceive them rather neutral with a small positive tendency which correlates with a manual classification of the tweets. We then used the KNN algorithm of Text Mining to classify the tweets according to the expressed risk perception. As the algorithm expects only one object (e. g. one tweet) as input, a script has been implemented which iterates through all tweets. Applying this algorithm to the training data set resulted in more than half of the tweets being classified "neutral". While 89 % of the tweets have been classified correctly, about 384 tweets could not be classified at all.

Next, we applied the PAL algorithm Naive Bayes. It was executable but classified more than 99 % of the tweets as neutral. Considerable improvement would be needed to get better results.

So the result was not very meaningful. Applying the PAL algorithm C4.5 Decision Tree is not really applicable for analyzing tweets as it expects several columns for input while the tweets are only stored in one column.

We only had some preliminary results using R as we were facing issues with embedding additional libraries and working on the tweets, for example because of the length of the tweets. For the Entity and Fact Extraction, R is not in the focus. In the previous project phase, we needed R for generating the document-term-matrix. With the new version of SAP HANA (SPS12 instead of SPS10), it was possible to generate it within SAP HANA Therefore, we did not proceed with exploring the possibilities of R in the current project phase.

[2] For further information, please read the project report of our former project at HPI Future SOC Lab (Spring 2016) [1].

3 Project Goal

The goal of this project was to build a prototype web application with a visual representation of the perception and dissemination of risks and benefits on Twitter. This application shall offer a quick and easy analysis to risk managers in near real-time.

4 Project Design

4.1 System Configuration

We were provided with a SAP HANA SPS12 with 1 TB RAM and 32 Cores (CPU). Additionally, we used the SAP HANA PAL and had access to an R Server. We used SAP HANA Studio, version 2.0.11, which is Eclipse-based, the XSJS Engine, the SAPUI5 development framework (v1.28.28), the SAP HANA TwitterAdapter for data provisioning, and OData Services for accessing the Data.

Additionally, we implemented an external Java application, which is the only non-native SAP HANA component. Fig. 1 gives an overview of our application architecture.

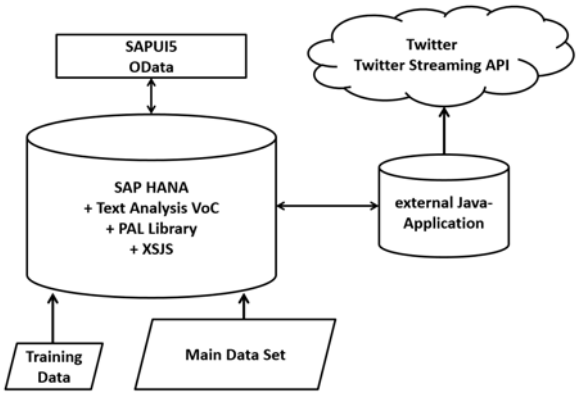


Figure 1: Application Architecture
(Source: own illustration)

4.2 Dataset

We used three datasets which we collected in previous work: a training data set for evaluation and configuring the algorithms, a more extensive dataset for applying the algorithms, and near real-time tweets gathered

from the Twitter Streaming API. While the training data has been the same as in the previous project, the extensive data has been expanded in the meantime.

The training data provided in a csv-file comprised 7,482 Tweets about autonomous driving, which have been classified manually, so that they could be used to train the algorithms. They have been assigned to the classes "benefit" (750 Tweets), "neutral" (6000 Tweets), and "risk" (750 Tweets). When applying the algorithm on these, the quality of the results can be judged by comparing the actual classification of the Tweets with the results of the algorithms applied.

The extended dataset comprised more than 1.8 million of tweets (not classified).

4.3 Data Pre-processing

Data pre-processing was done by applying means of the previous project to the Twitter dataset as well as new methods. Thus, URLs, empty tweets and duplicates were deleted. Twitter-specific elements like hashtags were marked. Off-topic tweets were filtered out.

4.4 Entity and Fact Extraction

In this project, we built on the results of our previous project and concentrated on the use of the native SAP HANA algorithm Text Analysis (VoC). Therefore, a customized fulltext index, which is needed for Entity and Fact extraction, has been created. This includes the steps configuration, dictionaries and extraction rules. The configuration is based on the "EXTRACTION_CORE_VOICEOFCUSTOMER" provided by SAP HANA. This contains default configurations and the linkage to dictionaries and extraction rules. We used two dictionaries: The first one is based on the English thesaurus with some modifications with regards to the topic of autonomous driving. Entity types such as "minor problem" or "major problem" already existed, but not for "risk", "benefit", and "neutral". Therefore, we implemented them and the second dictionary now contains new entity types for autonomous driving: "risk", "neutral", and "benefit". The extraction rules include rules for complex entity types to support risk and benefit identification. All customized dictionaries and extraction rules have been created based on the training dataset using variations of the document-term-matrix generated by SAP HANA Text Mining.

The algorithms used previously classified tweets as a whole. In contrast to this, our customized entity and fact extraction allows to analyse tweets on a sub-sentence level, which is the same as with SAP HANA Voice of Customer analysis. The results are satisfying considering the use of manual methods for developing dictionaries and extraction rules. Applying the customized entity and fact extraction to the training data, 69 % of the risks and 41 % of the benefits have been detected. As in the training dataset, tweets are classified and not tokens, we are not able to give meaningful information about the false-positive-rate. For the extensive dataset, the allocation for risks, neutrals and benefits is 16 %, 79 %, and 5 %. This result matches the observed allocation of the training dataset and as well the results of the previous project (10 %, 80 % and 10 %) considering the complexity of benefit detection and the increased number of risks detected due to the sub-sentence analysis.

4.5 Web Application for Visualization

For visualizing the results of the classification of risks and benefits, we built a web application based on SAP HANA. This web application permanently executes updates for being able to constantly display changes of the user behaviour, e.g., to show a sudden increase of perceived risks. The user interface consists of two tabs: The first tab (cp. fig. 2) shows the classified risks and benefits as perceived by the tweet authors (word cloud), the temporal (area chart), and the geographical (world map) dissemination as well as risk amplifying and risk attenuating Twitter users (network graph). On the second tab (cp. fig. 3), the raw tweets are listed within a table.

Figure 2: First tab of the web application
(Source: Screenshot from prototype)

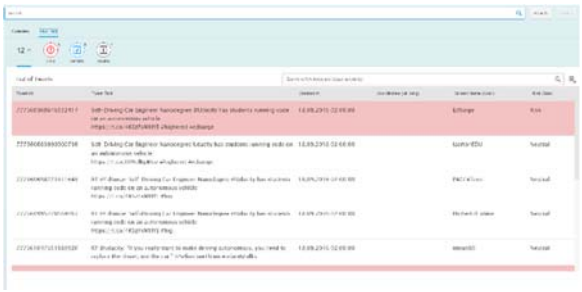


Figure 3: Second tab of the web application
(Source: Screenshot from prototype)

In order to implement the web application, we used the native SAPUI5 development framework and the XSJS engine. Our SAPUI5 version did not include the sap.viz framework for visualizing data using charts. Since the latest SAPUI5 version available (v1.38.8) has this framework deprecated [8], we decided to use external JavaScript libraries (vis.js [9], Highcharts

[10], Highmaps [11], and jQCloud [12]) for the visualization.

Our web application allows the user to search for tweets in near real-time using the Twitter Streaming API. It triggers the analysis of the tweets and visualizes the results.

With the web application, we first tried to gather tweets using the SAP HANA data provisioning agent (dpagent) using the predefined "TwitterAdapter". Unfortunately, the dpagent turned out to be not very reliable, since it frequently stopped working after a short period of time each time was restarted. Our workaround is a self-developed Java application implementing Twitter4J [13]. This is running on a virtual machine (VM), and is not a native SAP HANA element. The Java application polls SAP HANA for the topics the users submitted on the web interface of the application and which were saved within a SAP HANA table after submit. The Java application then filters the Twitter Streaming API for these topics. Next, using XSJS, the resulting tweets are persisted in SAP HANA tables and the analysis of them is triggered. The results of the analysis are frequently pulled by the web application via OData services and then visualized (cp. Fig. 4). A minor problem, which we have not been able to solve yet was the unreliability of the connection to the VM using SSH in order to deploy and manage the Java application.

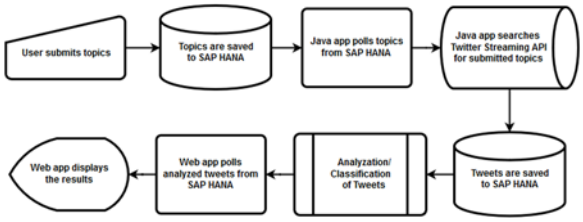


Figure 4: Flowchart of the web application

5 Conclusions and Outlook

As with Entity and Fact Extraction, we were able to get acceptable results. However, further improvements can be applied in the future: Problems resulted from rare documentation for customizing Entity and Fact Extraction. The most important source of information were the SAP reference guides [14, 15, 16]. For customizing the extraction rules, it is recommended to study the existing files and then modify a copy of them, rather than starting a new and empty file [17]. Within the SAP HANA (SPS12), existing dictionaries or extraction rules were not accessible – and therefore could not be adjusted. For the English thesaurus, we could access an older version of SAP HANA, so for us, this problem was limited to extraction rules. Additionally, debugging of dictionaries and extraction rules is difficult as errors are handled only after activation. Short error messages only show the line number. The scope of our extraction rules is limited by the limited training dataset which has been used to create the dictionaries and extraction rules. Applying a larger training dataset or even the use of machine-learning would be suitable to expire the scope.

However, the current extraction leads to satisfying results regarding risk and benefit detection. Although the results of token-based analysis are hardly comparable to those of other algorithms, the Entity and Fact Extraction provides a suitable set of results in the case of autonomous driving.

Since the Entity and Fact Extraction was developed using data regarding autonomous driving, the web application prototype, which uses the results of the Entity and Fact Extraction, is currently optimized for use cases regarding autonomous driving. However, the resulting prototype adequately visualizes the perception and dissemination of risks and benefits on Twitter in near real-time. The system provided allowed us to develop the prototype and to implement the whole process from gathering and analyzing near real-time data to the visualization of the results. Except the call of the Twitter Streaming API, everything could be accomplished using native SAP HANA means.

References

[1] M. Knigge, C. Kohl, G. Baader, H. Kienegger, H. Krcmar: *Sentiment Analysis on Twitter Data Using R Algorithms*. In press.

[2] *Busfahrer verwirrt Google-Auto – da kracht's*. Cited 06.10.2016. http://www.manager-magazin.de/unternehmen/autoindustrie/google-selbstfahrendes-auto-verursacht-erstmals-unfall-a-1079968.html, März 2016.

[3] L. Reiche: *Tesla unter "Autopilot" rammt Reisebus auf Autobahn*. Cited 06.10.2016. http://www.manager-magazin.de/unternehmen/autoindustrie/tesla-model-s-unfall-mit-autopilot-bus-auf-24-gerammt-a-1114664.html, September 2016.

[4] P. Russom: *Big Data Analytics*. TDWI Best Practices Report, Fourth Quarter, 1-35, 2011.

[5] L. Klie: *Listening to the Voice of the Customer*. In: CRM Magazine, Vol. 16, No. 1, 2012.

[6] C. Ware: *Foundation for a Science of Data Visualization*. In: Information Visualization: Perception for Design, Eds. Elsevier Science, 1-27, 2004.

[7] E.R. Tufte: *Visual Explanations: Images and Quantities, Evidence and Narrative*, Graphics Press, 1997.

[8] SAPUI5: Cited 09.10.2016. https://sapui5.hana.ondemand.com/#docs/api/symbols/sap.viz.ui5.core.BaseChart.html, 2016.

[9] B.V. Almende: *vis.js*. Cited 21.08.2016. http://visjs.org, 2016.

[10] Highsoft AS: *Highcharts*. Cited 21.08.2016. http://www.highcharts.com/products/highcharts, 2016.

[11] Highsoft AS: *Highmaps*. Cited 21.08.2016. http://www.highcharts.com/products/highmaps, 2016.

[12] L. Ongaro: *jQCloud*. Cited 21.08.2016. http://mistic100.github.io/jQCloud, 2016.

[13] Y. Yamamoto: *Twitter4J*. Cited 21.08.2016. http://twitter4j.org/en/index.html, 2016.

[14] [H] SAP: *SAP HANA Text Analysis Extraction Customizing Guide*. Cited 10.10.2016. http://help.sap.com/saphelp_hanaplatform/helpdata/en/20/31dfe5e9754d0fb09b5ca24fd0329f/frameset.htm, 2014.

[15] [I] SAP: *SAP HANA Text Analysis Developer Guide*. Cited 10.10.2016. http://help.sap.com/hana/SAP_HANA_Text_Analysis_Develper_Guide_en.pdf, 2016.

[16] [J] SAP: *SAP HANA Text Mining Developer Guide*. Cited 10.10.2016. http://help.sap.com/hana/SAP_HANA/Text_Mining_Developer_Guide_en.pdf, 2016.

[17] [K] A. Waite, Y. Meessen, B. Miller, M. Wiesner: *Text Analytics with SAP HANA Platform*. Cited 23.04.2016. https://open.sap.com/files/82000803-f0d5-41c5-ab68-5495b5437895, 2015.

Advanced Dynamic Evolutionary Computing Using SAP HANA

Wei Cheng
SAP Innovation Center Network, Potsdam, Germany
wei.cheng@sap.com

Julia Jordan, Bernd Scheuermann
Hochschule Karlsruhe, University of Applied Sciences, Karlsruhe, Germany
bernd.scheuermann@hs-karlsruhe.de

Abstract

This paper reports on an approach to advanced evolutionary optimization leveraging SAP HANA to expedite the search process subject to change events arising at runtime. The implemented system exploits optimization knowledge persisted on HANA serving as associative memory to better guide the optimizer through changing environments. For this, specific strategies for knowledge processing, extraction and injection have been developed and evaluated. Moreover, prediction methods provided by PAL have been embedded, and empirical results indicate that they can suitably anticipate forthcoming dynamic change events based on the evaluation of historical records of previous change events and of optimization knowledge managed by HANA.

1 Introduction

For decades Evolutionary Algorithms [8] have been established heuristics to tackle NP-hard optimization problems which are inherent to countless industrial applications. Such problems include, e.g., the Vehicle Routing Problem (VRP), the Traveling Salesperson Problem (TSP), the Knapsack Problem or many other problems in production, warehouse and transportation logistics. Typically, the search for good solutions to such problems can consume up to several hours or even days. The hitherto best solution found, like a transport plan or a production schedule, would then be used for planning and executing logistics operations. In practice, however, several aspects like the objective function, the size of the problem instance or constraints may be subject to changes, either during optimization, or maybe later during logistics operations. In such cases, the optimization problem is called *dynamic* and previously good solutions might have become inferior, or previously bad solutions might turn superior. Therefore it is essential that every relevant change of the optimization problem is taken into account. However, calculation time is commonly restricted and one cannot afford to restart optimization from scratch. Instead it is advisable to exploit existing optimization knowledge from the running optimization so as to quickly react to and to recover from dynamic changes arriving.

This paper reports on the implementation and on the empirical evaluation of an Evolutionary Algorithm which interfaces with SAP HANA and exploits its strengths to expedite the search process in dynamically changing environments. It is shown, how SAP HANA can be used as a knowledge store that embodies an associative memory to the optimization algorithm. Such knowledge includes, e.g., historical logs of visited search areas, environmental data, and recorded change events. In contrast to previous work, it is examined in how far in-memory database technology can help increase and manage the amount of stored knowledge in order to better guide the optimization process. Furthermore, SAP HANA is employed as a storage for predictive knowledge that is accessed and analyzed through PAL [16] to make the optimizer better prepared for prospected changes, to quickly respond to such changes and to easier recover from their impact.

The remainder of the paper is structured as follows: Section 2 briefly introduces to dynamic evolutionary computing. Section 3 outlines the architecture and the methods of the HANA-based system implemented. Subsequently, a sample of results from extensive experiments carried out in the Future SOC Lab are presented in Section 4. Concluding remarks are provided in Section 5.

2 Dynamic Evolutionary Optimization

Prior to introducing the Evolutionary Algorithm, the static Knapsack Problem shall be defined and henceforth be used to exemplify the strategies proposed in this paper. In its static variant, the 0/1 Knapsack Problem [12] is described by a set of n items of weight w_j and value v_j where $j \in \{1, \ldots, n\}$. A candidate solution $X = (x_1, \ldots, x_n)$ problem represents a subset of all items, with $x_j \in \{0, 1\}$ indicating if item j is included in the knapsack which has a capacity of C. The goal is to maximize the total value of items included in the knapsack such that the sum of their weights is less or equal to the knapsack capacity: Maximize

$$f(X) = \sum_{j=1}^{n} v_j x_j,$$

subject to

$$\sum_{j=1}^{n} w_j x_j \leq C, x_j \in \{0, 1\}.$$

As the Knapsack Problem is known to be $NP - hard$, evolutionary algorithms [8] are one common heuristic to search for near optimal solutions. Inspired by the principles of natural evolution, the main idea behind evolutionary optimization is to represent solutions of an optimization problem as a set of individuals called population. The size of the population shall be denoted as p. An individual $i \in \{1, \ldots, p\}$ is encoded in a chromosome X_i representing the individual's genotype. In the case of the Knapsack Problem, individual i is encoded as n-bit chromosome $X_i = (x_{1i}, \ldots x_{ni})$ with $x_{ji} \in \{0, 1\}$, where $x_{ji} = 1$ means that item number j is contained in the knapsack, and $x_{ji} = 0$ otherwise. The Evolutionary Algorithm aims to incrementally improve on the set of individuals by mimicking the principles of natural selection, recombination, mutation and survival of the fittest (cf. [8] for a detailed introduction).
Accordingly, the dynamic knapsack problem extends its static counterpart by introducing time-dependent variance: capacity $C(t)$, weights $w_j(t)$ and values $v_j(t)$ are considered as dynamic over time t. The goal is to maximize

$$f(X, t) = \sum_{j=1}^{n} v_j(t) x_j$$

at any time t, subject to

$$\sum_{j=1}^{n} w_j(t) x_j \leq C(t), x_j \in \{0, 1\}.$$

For dynamic optimization, one is interested in reusing existing optimization knowledge to quickly react to and to recover from dynamic changes to the environment. This implies that algorithms in dynamic environments are no longer focused on locating a stationary optimal solution. Rather they need to remain flexible to be able to track the movement of peaks through space and time.
Related work started with early contributions by Fogel at al. [7], evolutionary algorithms have ever since been the most common approach to solving dynamic optimization problems [4]. Rohlfshagen [15] defined a problem based on a class of 0/1 dynamic knapsack problem, which are generated by a small set of real-valued parameters. Branke et al. [3] analyzed different representations on dynamic multi-dimensional knapsack problem. Simões [17] applied diversity-maintaining techniques and memory strategies to evolutionary algorithms for the dynamic knapsack problem.

3 Dynamic Evolutionary Optimization Using SAP HANA

3.1 Overview

Figure 1 visualizes the architecture of the implemented system. The core algorithm, based on the

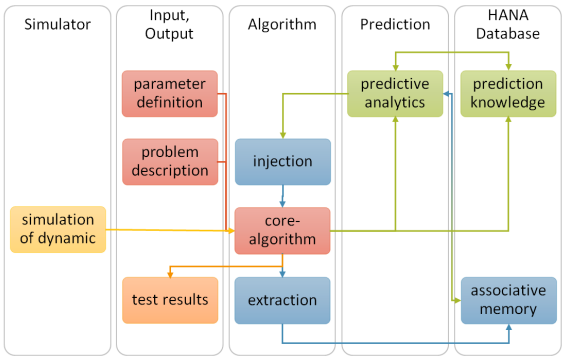


Figure 1: Architectural overview.

standard EA approach introduced in the previous section, is started reading a set of parameters and the problem definition. Since the problem is considered dynamic, the simulator continuously adapts this problem definition, whereupon the changes are propagated to the core algorithm. Moreover, in dynamic environments, it is required to augment the algorithm with mechanisms to preserve solution diversity thereby preventing premature convergence. A set of individuals are perpetually extracted from the current population and persisted into the associative memory held in SAP HANA. This database is queried for a number of suitable individuals which are injected into the core algorithm whenever the

problem definition changes. The choice of individuals to be injected may be guided through prediction knowledge stored on SAP HANA. This knowledge is analyzed exploiting the routines provided by PAL [16] to predict and to better prepare for forthcoming dynamic changes.

3.2 Simulator

An environment $e(t)$ is considered to represent the definition of the optimization problem at time t, where the time is supposed to be the generation number. The environment is constituted by a tuple of problem parameters that are subject to change. In the case of the knapsack problem, the environment $e(t) = (C(t), v(t), w(t))$ shall be signified by the knapsack capacity $C(t)$, its item weights $w(t) = (w_1(t), \ldots, w_n(t))$ and item values $v(t) = (v_1(t), \ldots, v_n(t))$. The simulator is implemented to change the environment at a constant frequency of T generations. Furthermore, it is assumed that the environment changes in a cyclic manner at a cycle length of L. Hence any environment will recur every $L \cdot T$ generations as visualized in Figure 2.

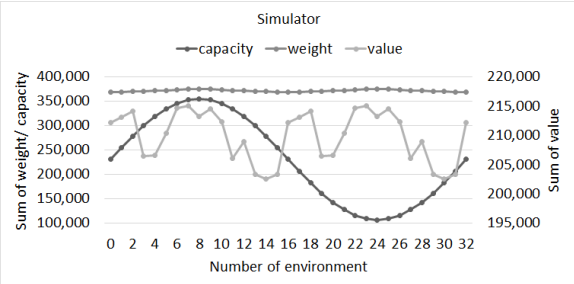


Figure 2: Sample scenario simulating a dynamic knapsack problem with cycle length $L = 32$.

3.3 Database Coupling

The application has been implemented in server-side JavaScript using the FutureSOCLab infrastructure. Figure 3 outlines the coupling with the SAP HANA database. Sequences of experiments are scheduled through an XS Job. Each experiment is identified by an ID which is also used to query the database for the input parameters, including, e.g., the problem definition and the algorithms parameters. At runtime the algorithm continuously logs the calculation results in terms of generation numbers, fitness value statistics, best solutions, time stamps and further performance indicators. Additionally optimization knowledge is maintained in dedicated associative and predictive knowledge stores, respectively, also residing on HANA.

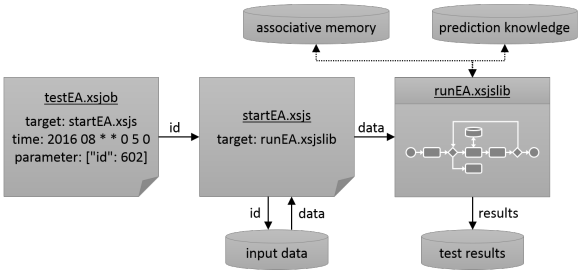


Figure 3: Interfacing the SAP HANA database

3.4 Fitness Function

The fitness of an individual represents it solution quality. This also includes a penalty cost term assigned to individuals representing overfilled knapsacks. Hence, the fitness of individual i is calculated as

$$fitness\,(i) = \sum_{j=1}^{n} (v_j \cdot x_{i,j}) \cdot \left(1 - \max\left(0; \frac{\sum_{j=1}^{n} (w_j \cdot x_{i,j}) - C}{C}\right)^{\lambda}\right)$$

with parameter λ set to 0.5.

3.5 Diversity Management

Diversity management is an essential factor in dynamic evolutionary optimization, because a diverse population can better react to a change than a converged population [5, 1]. *Fitness Sharing* [9] is probably the most commonly used diversity management techniques. The shared $fitness^*(i)$ of individual i depends on its distance to other individuals k:

$$fitness^*\,(i) = \frac{fitness\,(i)}{\sum_{k=1}^{p} sh\,(d\,(i,k))}$$

with

$$sh\,(d\,(i,k)) = 1 - \left(\frac{d\,(i,k)}{\sigma_s}\right)^{\alpha}$$

if $d\,(i,k) < \sigma_s$, 0 otherwise , where $d\,(i,k)$ is the hamming distance between individual i and k. Parameter p is the size of population (further choosing $p = 40$ throughout this paper). Parameter α is a constant which defines the shape of sharing function, and is set to *one* according to [17]. Niche radius $\sigma_s = 257.6$ has been calculated according to [6]. Further strategies in this paper are *Deterministic Crowding Selection (DCS)* [13] and *Mating Restricted Tournament (MRT)* [11].

3.6 Associative Memory

An associative memory needs to address four main issues [14]: (1) how to organize the memory, (2) when to *extract* which individuals from the EA to the memory (3) how to *update* the memory and (4) when to *inject* individuals from the memory to the EA, see Figure 4 and blue boxes in Figure 1.

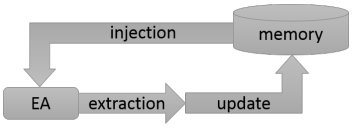


Figure 4: Extraction, update of memory and injection

The associative memory is organized in database tables on SAP HANA. Table GENOTYPES stores good old solutions, the other tables hold environmental information. Figure 5 illustrates an example. When an old environment reappears, the environmental information from the memory is used to identify good solutions which had previously been successful in this environment. These individuals are then injected into the EA.

CAPACITY

TEST-ID	ENV-ID	CAP
String	Integer	Double
602-1-6-376	0	231001,0000
602-1-6-376	1	255267,3097

WEIGHT

TEST-ID	ENV-ID	WEIGHT001	...	WEIGHT500
String	Integer	Double		Double
602-1-6-376	0	533,000		852,000
602-1-6-376	1	599,625		852,000

VALUE

TEST-ID	ENV-ID	VALUE001	...	VALUE500
String	Integer	Double		Double
602-1-6-376	0	276,000		103,000
602-1-6-376	1	299,544		103,000

environmental information

old solutions/ individuals

GENOTYPES

TEST-ID	ENV-ID	GENERATION	INDIVIDUAL	GENE001	...	GENE500
String	Integer	Integer	Integer	Integer		Integer
602-1-6-376	0	10000	8	1		0
602-1-6-376	0	140000	17	1		1

Figure 5: Database schema of the associative memory

The *extraction* takes place at equally spaced intervals. Grefenstette and Ramsey [10] recommend to substitute 50 % of the population with individuals from the memory after a change, in order to have enough individuals in the memory, the number of extracted individuals is 50 % of the population size. To decide which individuals to extract an *importance value* [2] is calcuated for each individual i as

$$\begin{aligned} imp\left(i\right) &= \gamma_f \cdot imp_{fit}\left(i\right) \\ &\quad +\gamma_d \cdot imp_{div}\left(i\right) \\ &\quad +\gamma_a \cdot imp_{age}\left(i\right) \end{aligned}$$

with $\gamma \in [0,1]$ and $\gamma_f + \gamma_d + \gamma_a = 1$. The terms $imp_{fit}\left(i\right)$, $imp_{div}\left(i\right)$ and $imp_{age}\left(i\right)$ express the indiviual's relative contribution to the fitness, diversity and age of the population. They are computed as follows:

$$imp_{fit}\left(i\right) = \frac{fitness\left(i\right)}{\sum_{k=1}^{p} fitness\left(k\right)}, \tag{1}$$

$$imp_{div}\left(i\right) = \frac{\sum_{s=1}^{p} d\left(i,s\right)}{\sum_{z=1}^{p}\sum_{s=1}^{p} d\left(z,s\right)}, \tag{2}$$

$$imp_{age}\left(i\right) = \frac{age\left(i,g\right)}{\sum_{k=1}^{p} age\left(k,g\right)} \tag{3}$$

where $d\left(z,s\right)$ is the Hamming distance between individuals z and s. The age of an individual i is set to $age\left(i,g\right) = 0$ if the individual was created in generation g, and $age\left(i,g-1\right)+1$ otherwise. The population is sorted descending by importance and the 50 % with the highest importance value are extracted to be stored in the memory.
When the *extraction* takes place and the current environment does not exist in the HANA memory so far, the new environmental information is stored in HANA and individuals are inserted to the memory. If a similar environment exists in the memory, the extracted individuals will be used to *update* table $GENOTYPES$ (see Figure 5).
When the *injection* takes place (e.g. after a change), the environmental tables in the HANA memory are searched for an environment similar to the new environment. If the search is successful, the stored individuals from the memory will replace similar individuals in the population of the EA. Otherwise only immigrants will be generated to increase diversity.

3.7 Change Prediction

The associative memory component interacts with the predictive analytics component. Prediction is triggerd after each change and aims to anticipate the generation and nature of the next change. Therefor *prediction knowledge* on previous changes is stored in the HANA database (see Figure 1). Based on the previous changes, a stored procedure from SAP PAL for polynomial regression ($POLYNOMIALREGRESSION$) is used to calculate

- the anticipated generation of the next change,
- which of the parameters c, w_j and v_j are going to change and
- how they will change.

Other PAL procedures like *forecast smoothing*, *neural networks* or *auto regressive integrated moving average (ARIMA)* are potential candidates for predictive analytics as well.
Based on the output of the prediction the expected environment can be simulated and the associative memory is searched for individuals, which had been successful in an environment similar to the simulated one. If such individuals are found, they remain in a temporary buffer in order to be injected right before the anticipated change occurs. If no suitable individuals are found in the memory, immigrants will be inserted to the population when the next change occurs to increase diversity. Figure 6 illustrates an example for the calculation of the generation of the next change in a scenario where change occurs every 7000 generations. The

PR-INPUT-GEN		
ID	Y	X1
Integer	Integer	Integer
1	7000	1
2	14000	2
3	21000	3
4	28000	4
5	35000	5
6	42000	6

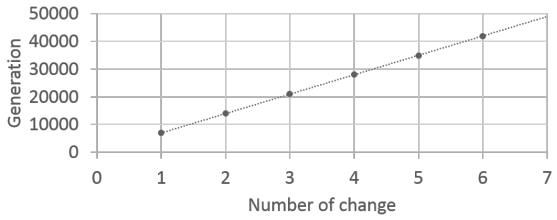


Figure 6: Prediction of the generation of the next change using regression

input table for the stored procedure is organized in the way required by PAL [16, p. 273]. The dotted line in the diagram illustrates the corresponding regression function, which is used to calculate the generation of the 7^{th} change.
The prediction accuracy is calculated as $acc = 1 - err$, where err will be one if the acctual change occurs too early or if no prediction was made at all. Otherwise err is calculated as $err = \max\left\{1, (err_{gen} + err_{out}) \cdot \frac{1}{2}\right\}$ where err_{gen} is the *relative error* of the predicted generation and err_{out} is the relative error of the predicted new value of either c, w_j or v_j. They are calculated as

$$err_{gen} = \left| \frac{gen_{prd} - gen_{chg}}{gen_{chg}} \right|, \qquad (4)$$

$$err_{out} = \left| \frac{cwv_{prd} - cwv_{chg}}{cwv_{chg}} \right|. \qquad (5)$$

prd stands for the predicted value and *chg* for the actual value when the change occurs. *cwv* is either the capacity C or the weight w_j or value v_j of one single item j in the knapsack. That means, for each item which has a predicted change, the accuracy is computed. It can then be averaged for further evaluation.

4 Evaluation Results

Extensive tests on the FutureSOCLab infrastructure were conducted to evaluate the performance of the implemented algorithm, the memory and the predictive component. This paper can only outline a fraction of the experiments and results obtained, see figure 7 and 8.

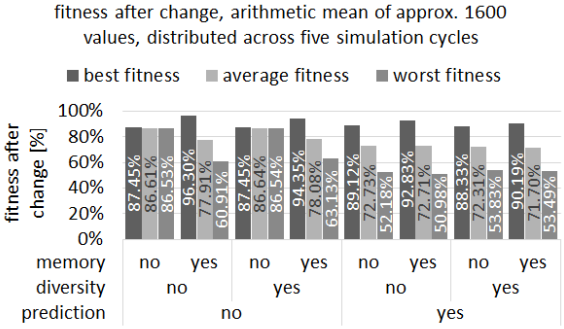


Figure 7: Test results. Impact on solution quality.

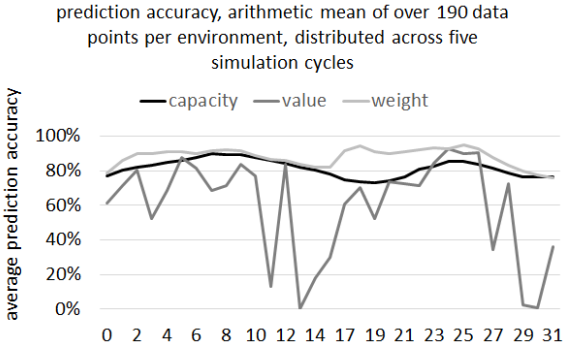


Figure 8: Test results. Prediction accuracy.

The diagrams show the decrease of the fitness after a change in percent and the prediction accuracy. The tests where conducted with and without associative memory as well as with and without prediction. Furthermore, the use of fitness sharing and mating restriction to maintain diversity throughout the run is evaluated, indicated as *diversity* in Figure **??**.
As shown in Figure 2 the simulator creates alternating changes in c (sine function) as well as fluctuating values v_j. Whereas the prediction accuracy of the capacity lies between 70 % and 90 %, the fluctuations modeled are nearly impossible to

predict, as confirmed by the low prediction accuracy in Figure 8. The low prediction accuracy is also the reason why the decrease of the best fitness after a change can not be lowered by the predictive component but is even increased. This can mainly be attributed to the hardness of predictability and not the predictive analysis itself. However, trying different prediction techniques might increase the performance of the predictive component. It is also important to consider the computation time for prediction, which consumes around 30 to 80 at each call.
The associative memory on HANA exerted a considerably positive impact on fitness. The drop of the fitness after a change is reduced to only 3.70 % compared to 12.55 % with no adaptation to dynamic environments at all (cf. Figure 7). The memory component requires only about 0.5 seconds for communication between algorithm and database when the memory is called.

5 Conclusion

This paper reported on an implemented approach to dynamic evolutionary optimization exploiting the strengths of in-memory computing. Empirical studies suggest that SAP HANA can enable the optimizer to learn from the decisions of the past and to make better informed decisions in the forthcoming iterations of the optimization algorithm. This positive effect of associative memory seems becomes particularly apparent in recurring environments. In such cases, the contribution of associative memory is strong. Using HANA allows storing and maintaining huge amounts of data on previously visited solutions. The test results also indicate, that there is a strong interdependence between the frequency of change and the extraction strategy, meaning that the interval for extraction needs to adapt to the frequency of change in order to ensure maximum efficiency of the associative memory.

References

[1] J. Branke. Memory enhanced evolutionary algorithms for changing optimization problems. In *Proc. of CEC 99*, 1999.

[2] J. Branke. Memory enhanced evolutionary algorithms for changing optimization problems. *Congress on evolutionary computation CEC99*, pages 1875–1882, 1999.

[3] J. Branke, M. Orbayı, and Ş. Uyar. *Applications of Evolutionary Computing: EvoWorkshops 2006*, chapter The Role of Representations in Dynamic Knapsack Problems, pages 764–775. Springer, 2006.

[4] C. Cruz, J. R. Gonzalez, and D. A. Pelta. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Comput.*, 15:1427–1448, July 2011.

[5] C. Cruz, J. R. González, and D. A. Pelta. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing*, 15(7):1427–1448, 2011.

[6] K. Deb and D. E. Goldberg. An investigation of niche and species formation in genetic function optimization. In *Proceedings of the 3rd ICGA*, pages 42–50, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[7] L. Fogel, A. Owens, and M. Walsh. *Artificial intelligence through simulated evolution*. Wiley, 1966.

[8] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[9] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Genetic Algorithms and Their Applications: Proceedings of the 2nd ICGA*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[10] J. J. Grefenstette and C. L. Ramsey. Case-based initialization of genetic algorithms. *Proceedings of the 5th ICGA*, pages 84–91, 1993.

[11] H. Ishibuchi and Y. Shibata. *Mating Scheme for Controlling the Diversity-Convergence Balance for Multiobjective Optimization*, pages 1259–1271. Springer, Berlin, Heidelberg, 2004.

[12] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*. Springer, 2004.

[13] S. W. Mahfoud. *Niching Methods for Genetic Algorithms*. PhD thesis, Champaign, IL, USA, 1995. UMI Order No. GAX95-43663.

[14] T. T. Nguyen, S. Yang, and J. Branke. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24, 2012.

[15] P. Rohlfshagen and X. Yao. The dynamic knapsack problem revisited: A new benchmark problem for dynamic combinatorial optimisation. In *Proc. of the Evo Workshops*. Springer, 2009.

[16] SAP. SAP HANA Predictive Analysis Library (PAL): SAP HANA Platform SPS 11, Document Version: 1.0 2015-11-25, 2015.

[17] A. Simões and E. Costa. *Artificial Neural Nets and Genetic Algorithms*, chapter An Immune System-Based Genetic Algorithm to Deal with Dynamic Environments: Diversity and Memory, pages 168–174. Springer Vienna, Vienna, 2003.

An early warning indicator for deception detection in social media

Estée van der Walt
Department of Computer Science
Security & Data Science Research Group
University of Pretoria, South Africa
estee.vanderwalt@gmail.com

Prof J.H.P. Eloff
Department of Computer Science
Security & Data Science Research Group
University of Pretoria, South Africa
eloff@cs.up.ac.za

Abstract

The detection of identity deception is important for a variety of reasons. The research at hand propose to focus on the protection of minors on big data platforms, like social media, as a use case for identity deception detection. The nature of social media has exacerbated the difficulty of detecting identity deception. Not only does the volume of data grow daily through the contribution of the public and IoT, but the velocity and variety of data increase as well. These characteristics of social media data necessitates an intelligent identity deception indicator to automate the detection of such deception.

1 Project idea

Identity deception historically has been focused on the psychological aspects around why people lie [1] [2] and what are some cues for deception [3]. Little has however been done so far on social media and identity deception detection. Social media is still new as it only started around 2006 with the evolution of the web as what was known as Web 2.0 [4].

Current identity deception research, in social media, cover a variety of use cases. Below are some examples:

- influencing outcomes, like political campaigns [5]
- enhancing or damaging the image of a company's brand [6] [7]
- spam activity [8]
- spreading fake news [9] [6]

The above mentioned research mostly focus on fake accounts or bots which have been autogenerated for the indicated purpose [7]. The identity of these accounts is usually fabricated. Further research also proposed an algorithm for differentiating bot accounts from human accounts [10]. One of the indicators of bot accounts, for example, is that the timing of bot account tweets has very low entropy, i.e. is very predictable.

Human accounts are however of particular interest for the research at hand. Minors requires protection from predators, like pedophiles [11]. Pedophiles will typically lie about who they are (their identity) to groom or approach a minor [12].

Current proposed methods and algorithms in identity deception have been found lacking for the following reasons:

- not intelligently highlighting outliers for further manual intervention
- not extendible to combine a variety of different types of variables to form a more successful indicator
- not real time to be proactive towards detection and taking note of historical events to ensure a moving average result

The closest research found to date to the research at hand had success in showing potential identity deception when combining the name of the user with the color of the background account image [13].

The research project has been divided into various processes discussed in more detail during previous research papers. The focus of this phase of the research was to understand the variables in the dataset at hand in more detail. Each variable was evaluated and some basic machine learning was applied to understand the variables' relevance to one another. This specific process is highlighted in green in figure 1.

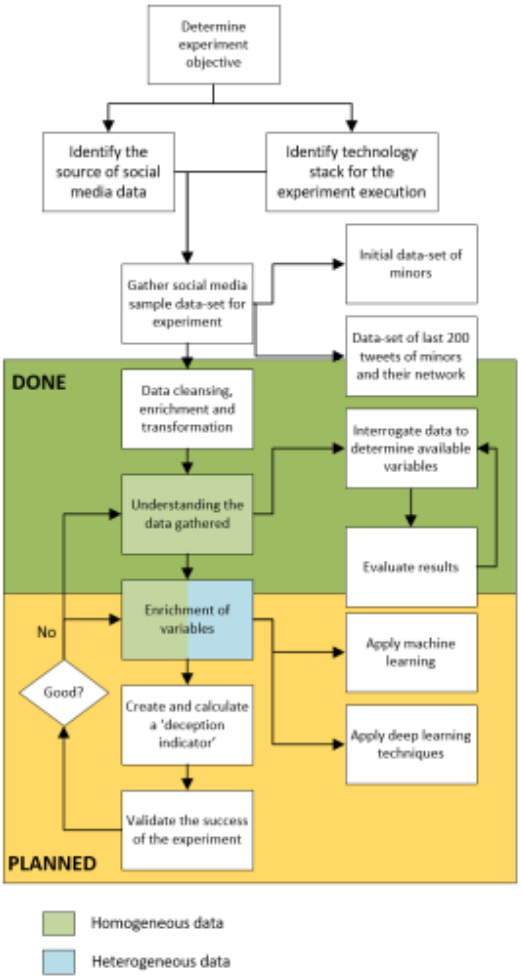


Figure 1: The project process diagram

1.1 Main deliverables

The main deliverables of the past six months were:

- To clean, enrich and transform the data.
- To understand the data through variable inspection.
- To explore machine learning algorithms for enrichment and addition of more variables to the research at hand.

2 Use of HPI Future SOC Lab resources

To reiterate past feedback, the following resources were used for the research at the HPI Future SOC lab:

- Twitter: The Twitter4j Java API was used to dump the data needed for the experiment in a big data repository.
- Hortonworks Hadoop 2.4: For the purposes of this experiment HDP Hadoop runs on an Ubuntu Linux virtual machine hosted in "The HPI Future SOC"- research lab in Potsdam, Germany. This machine contains 4TBs of storage, 8GB RAM, 4 x Intel Xeon CPU E5-2620 @2GHz and 2 cores per CPU. Hadoop is well known for handling heterogeneous data in a low-cost distributed environment, which is a requirement for the experiment at hand.

  Flume: Flume is used as one of the services offered in Hadoop to stream initial Twitter data into Hadoop and into SAP HANA.

  Ambari: For administration of the Hadoop instance and starting/stopping the services like Flume.

  Note that we have upgraded our Hadoop instance from version 2.3 to a stable version of 2.4.
- Java: Java is used to enrich the Twitter stream with additional information required for the experiment at hand and automate the data gathering process.
- SAP HANA: A SAP HANA instance is used which is hosted in "The HPI Future SOC"- research lab in Potsdam, Germany on a SUSE Linux operating system. The machine contains 4TBs of storage, 2TB of RAM (1.4TB effective) and 32CPUs / 100 cores. The in-memory high-performance processing capabilities of SAP HANA enables almost instantaneous results for analytics.

  The XS Engine from SAP HANA is used to accept streamed Tweets and populate the appropriate database tables.

- Machine learning APIs: Various tools are considered to perform classification, analysis and apply deep learning techniques on the data. These include the PAL library from SAP HANA, SciPy libraries in Python, Spark Mlib on Hadoop and the Hadoop Mahout service. For the research R and potentially Gephi to display graph data was the final choice. This decision was made due to support on these tools and libraries freely being available on the web community at a large scale.
- Visualization of the results will be performed by the libraries in R.

The following ancillary tools were used as part of the experiment:

- For connection to the FSOC lab we used the OpenVPN GUI as suggested by the lab.

- For connecting and configuration of the Linux VM instance we used Putty and WinSCP
- For connecting to the SAP HANA instance, we used SAP HANA Studio (Eclipse) 1.80.3

3 Findings in the Spring 2016 semester

The purpose of this phase of the research project was to clean and investigate the variables within the existing dataset. Basic cleanup was performed in R to remove any outliers.

Figure 2 shows the analysis of data where outliers have not been removed.

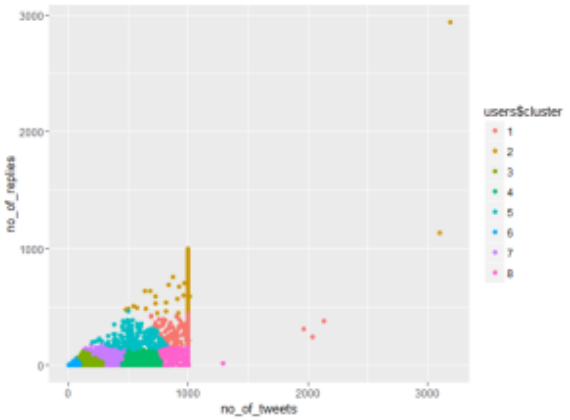


Figure 2: K-means clustering including outliers

The scaling of variables was also important in certain scenarios. It was found that certain machine learning algorithms performed better when the numerical ranges were in the same scale. For specific machine learning algorithms, like self-organized maps (SOM), scaling is a requirement.

Various machines learning algorithms where experimented with (all unsupervised):

- K-means clustering
- Hierarchical clustering
- Model based clustering
- Self-organized maps (SOM)
- Principal component analysis

These algorithms showed that the data could be categorized into the following main categories:

- Textual data, like the content of the tweet
- Numerical data, like the number of followers or retweets
- Images, like the account profile image
- GPS and time zone related data

The results of the algorithms also showed that GPS and time zone data influenced the other categories. Figure 3 shows example results highlighting that daily tweets patterns differ over time zones.

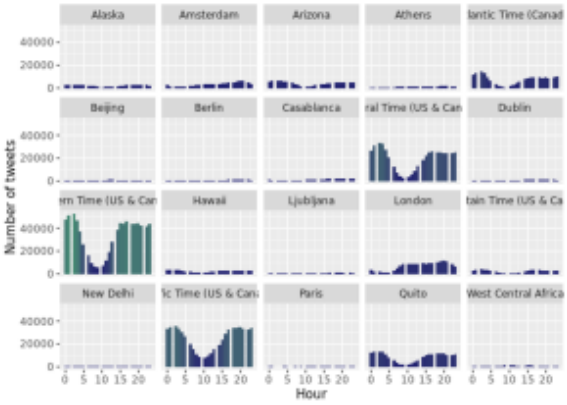


Figure 3: Tweet time per time zone per day

Based on these findings, GPS and time zone will not be regarded as a category of data going forward in our research but rather as an influencer or dimension of the others. Sentiment, for example, within the textual data category can differ per location and time zone.

The SAP HANA instance, virtual machine and storage was provided by the HPI FSOC research lab and the following is worth mentioning:

- There were no issues in connection.
- The lab was always responsive and helpful in handling any queries.
- The environment is very powerful and more than enough resources are available which makes the HPI FSOC research lab facilities ideal for the experiment at hand

We did experience issues with the memory on the SAP HANA server. This was found to be due to a large dataset being loaded as a row stored table into memory. The believe is that when this is changed to a column stored table, the issue will be resolved as only requested partitions of the data will be loaded into memory.

Overall we found that the environment and its power enabled the collection of a big dataset without issue. The support of the HPI FSOC research lab is appreciated.

4 Next steps for 2016/2017

The next steps in the project is to finalize the research project in the next semester.

The deliverables for this phase are:

- to finalize the experiments or hypothesis for the use case at hand
- to identify one indicator from each category of data as input to the identity deception indicator.
- To produce an identity deception indicator based on the identified variables per location and time zone
- To include tweet time and account open date as a factor for an average weight of the identity deception indicator
- To weight the different indicators, locations and time zones for the purpose of producing a more focused identity deception indicator
- To run the proposed model on real time data to identify outliers

References

[1] D. A. Kashy and B. M. DePaulo, "Who lies?," Journal of Personality and Social Psychology, vol. 70, p. 1037, 1996.

[2] G. Wang, H. Chen, and H. Atabakhsh, "Criminal identity deception and deception detection in law enforcement," Group Decision and Negotiation, vol. 13, pp. 111-127, 2004.

[3] B. M. DePaulo, J. J. Lindsay, B. E. Malone, L. Muhlenbruck, K. Charlton, and H. Cooper, "Cues to deception," Psychological bulletin, vol. 129, p. 74, 2003.

[4] K. Nath, S. Dhar, and S. Basishtha, "Web 1.0 to Web 3.0 - Evolution of the Web and its various challenges," in 2014 International Conference on Optimization, Reliabilty, and Information Technology (ICROIT), 2014, pp. 86-89.

[5] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic Deception Detection: Methods for Finding Fake," 2015.

[6] B. Drasch, J. Huber, S. Panz, and F. Probst, "Detecting Online Firestorms in Social Media," 2015.

[7] S. Gurajala, J. S. White, B. Hudson, and J. N. Matthews, "Fake Twitter accounts: Profile characteristics obtained using an activity-based pattern detection approach," in Proceedings of the 2015 International Conference on Social Media & Society, 2015, p. 9.

[8] S. J. Soman and S. Murugappan, "Detecting malicious tweets in trending topics using clustering and classification," in Recent Trends in Information Technology (ICRTIT), 2014 International Conference on, 2014, pp. 1-6.

[9] C. Chen, K. Wu, V. Srinivasan, and X. Zhang, "Battling the internet water army: Detection of hidden paid posters," in Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2013, pp. 116-120.

[10] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Who is tweeting on Twitter: human, bot, or cyborg?," in Proceedings of the 26th annual computer security applications conference, 2010, pp. 21-30.

[11] A. Schulz, E. Bergen, P. Schuhmann, J. Hoyer, and P. Santtila, "Online Sexual Solicitation of Minors How Often and between Whom Does It Occur?," Journal of Research in Crime and Delinquency, p. 0022427815599426, 2015.

[12] D. Bogdanova, P. Rosso, and T. Solorio, "Exploring high-level features for detecting cyberpedophilia," Computer Speech & Language, vol. 28, pp. 108-120, 2014.

[13] J. S. Alowibdi, U. A. Buy, S. Y. Philip, S. Ghani, and M. Mokbel, "Deception detection in Twitter," Social Network Analysis and Mining, vol. 5, pp. 1-16, 2015.

Resource Allocation Strategies for Elastic Data Stream Management Systems

Thomas Heinze[1,3], Zbigniew Jerzak[2], Christof Fetzer[3]

[1]SAP SE
Robert-Bosch-Strasse 30/34
69190 Walldorf, Germany
thomas.heinze@sap.com

[2]SAP SE
Rosenthaler Str. 30
10178 Berlin, Germany
zbigniew.jerzak@sap.com

[3] TU Dresden
Noethnitzer Str. 46
01187 Dresden, Germany
christof.fetzer@tu-dresden.de

Abstract

Elastic scaling allows cloud-based data management systems to handle unpredictable load changes by dynamically adding or removing resources. Dynamic resource (de)allocation increases the system utilization and reduces the operational cost. In this proposal we perform a large-scale evaluation of costs and SLAs in elastic data stream management systems. In our evaluation we focus on strategies, which decide where the load is moved.

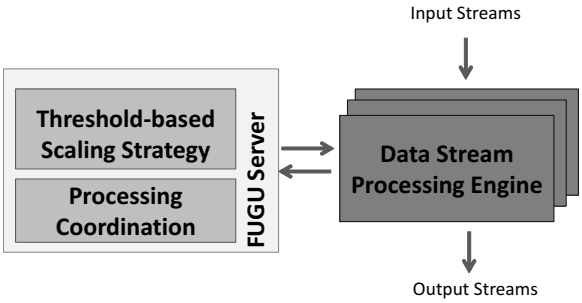


Figure 1: Architecture of FUGU

1 Introduction

Due to a constantly changing workload the utilization of cloud-based systems constantly varies. The utilization of a typical cloud-based system rarely exceeds 30% [11]. The major goal of all providers of cloud-based systems is to maximize their utilization while guaranteeing service level agreements (SLAs) for end users. Maximizing the utilization can be achieved by dynamically allocating and de-allocating resources (hosts). However, the higher the utilization of a given system the more difficult it is for such a system to fulfill user specified SLAs, such as latency and throughput guarantees. This fundamental trade-off is the main motivation driving the research behind the elastic scaling of data management systems.

In our current research we focus on elastic scaling of data stream management systems [7, 9] and publish/subscribe systems [4]. Our concepts have been implemented within a prototype, which uses different scaling as well as optimization techniques in order to achieve the best trade-off between utilization and user-specified SLAs. Our previous research focused on identifying the right scaling policies and determine which operators to move. We like to use the opportunity offered by the HPI Future SOC Lab to study an open problem in our system: the question on which host to place the moved operators. We used simplistic heuristics for this purpose so far [7, 9]. We started this activity in the HPI Future SOC Lab in Fall 2015, where we worked on our setup and presented some early results.

In this report we describe the results, which we achieved in context of the HPI Future SOC Lab Spring 2016. First, we describe the architecture of our prototype in more detail in Section 2. The operator placement problem is introduced in Section 3 and some evaluation results are presented in Section 4. Finally, we describe some conclusions in Section 5.

2 Background

The concepts presented here are implemented as an extension of the elastic data stream management prototype FUGU [7, 8] (see Figure 1). The existing system consists of a centralized management component, which dynamically allocates a varying number of hosts. The manager executes on top of a distributed data stream management engine, which is based on the Borealis semantic [1].

The data stream management system processes continuous queries, which can be modeled as directed acyclic graphs of operators. Our system supports primitive relational algebra operators (selection, projection, join, and aggregation) as well as additional data stream processing specific operators (sequence, source, and sink). Each operator can be executed on an arbitrary host and a query can be partitioned over

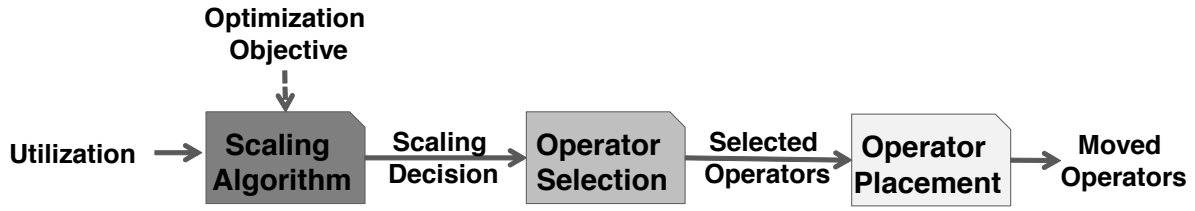


Figure 2: Scaling Strategy of FUGU

multiple hosts. The number of hosts is variable and dynamically adapted by the management component to changing resource requirements.

The centralized management component serves two major purposes: (1) it derives scaling decisions, including decisions on allocating new hosts or releasing existing hosts, and assigns operators to hosts; and (2) it coordinates the construction of the operator network in the distributed data stream management engine.

The management component constantly receives statistics from all running operators in the system. Based on these measurements and a set of thresholds and parameters, it decides when to scale and where to move operators. Typically, these thresholds and parameters are manually specified by the user. Our system supports the movement of both stateful (join and aggregation) and stateless operators (selection, sink, and source). A state of the art movement protocol [7, 12] ensures an operator moves to the new host without information loss.

2.1 Threshold-based Elastic Scaling

The scaling approach used by the FUGU server is illustrated in Figure 2. A vector of node utilization measurements (CPU, memory, and network consumption) and a vector of operator utilizations are used as input to the *Scaling Algorithm*. The *Scaling Algorithm* derives decisions that mark a host as overloaded or the system as underloaded. The *Operator Selection* algorithm decides which operators to move and the *Operator Placement* algorithm determines where to move these operators.

The default scaling strategy of FUGU is threshold-based, namely, a set of threshold rules are used to define when the system needs to scale. These thresholds mark either the entire system or an individual host as over/underloaded. A threshold rule describes an exceptional condition for the consumption of one major system resource (CPU, network, or memory), which triggers a scaling decision in FUGU. Some examples for these rules include:

1. A host is marked as overloaded if the CPU utilization of the host is above 80% for three seconds.

2. A host is marked as underloaded if the CPU utilization of the host is below 30% for five seconds.

The threshold-based rules need to be used carefully [6]. In particular, the frequent alternating allocation and deallocation of virtual machines, called thrashing, should be prevented. Several steps are taken in FUGU to avoid thrashing. First of all, each threshold needs to be exceeded for a certain number of consecutive measurements before a violation is reported. This number is called the *threshold duration*. In addition, after a threshold violation is reported, no additional scaling actions are done for the corresponding host for a certain time interval called a *grace period* (or cool-down time).

The load in a data stream management system is partitioned among all operator instances running in the system. Therefore, each scaling decision needs to be translated into a set of moved operators. The first problem is to identify which operators to move. This identification is done by the *Operator Selection* algorithm. If the system is marked as underloaded, it selects all operators running on the least loaded hosts. For an overloaded host, the *Operator Selection* algorithm chooses a subset of operators to move in a way, that the summed load remaining on the host is smaller than the given threshold. FUGU models this decision as a *subset sum problem* [10], where the operators on the host are the possible items and the threshold represents the maximum sum. We use a heuristic, which identifies the subset of all operator instances whose accumulated load is smaller than the threshold and no other subset with a larger accumulated load fulfilling this condition exists. All operators selected by this algorithm are kept on the host; the remaining operators are selected for movement.

The selected operators are the input of the *Operator Placement* algorithm, which decides *where* the operators should be moved. An operator can only be moved to a host, if the host has enough remaining CPU, network and memory capacity. The used heuristics can try to fulfill different objectives as discussed in the next section.

3 Operator Placement

The primary task of the operator placement is to assign operators to hosts in a way that the total number of hosts is minimized. Bin packing algorithms [5] are a well known solution to achieve this objective. A bin packing algorithm searches for an assignment of a set

of items to a set of bins. Each item has a weight and each bin has a capacity. The goal of a bin packing algorithm is to assign each item to exactly one bin in a way that (1) the number of bins is minimized and (2) the sum of the weights of all assigned items is smaller than the capacity of the bin. In the context of FUGU, an operator represents an item and its CPU usage its weight. A host is modeled as a bin with its CPU resource as the capacity. In addition, we use network and memory consumption as sub-constraints.
We implemented three well-known bin packing methods to study their performance for our problem:

FirstFit iterates over all available hosts based on the host ID, starting with host 1. An operator is placed on the first found host with enough capacity.

BestFit always studies all available hosts before placing an operator. The operator is placed on the host, which has enough capacity and the largest utilization of all hosts with enough capacity. This approach should minimize the unused capacity on all used hosts.

WorstFit always studies all available hosts before placing an operator. The operator is placed on the host, which has enough capacity and the smallest utilization of all hosts with enough capacity. This approach tries to achieve a balanced load between all used hosts.

These heuristics are well-studied and known for their good performance in terms of minimizing the number of used hosts. However, the problem for our elastic operator placement is slightly different, because the operator placement is executed not only once, but each time an overload or underload is detected. Therefore, the derived decision might be a good solution for the current situation, but can result in some drawbacks for later decisions. In our experiments, we observed, that using the *BestFit* heuristic increases the probability of overloaded hosts, because operators are always moved to hosts with already high load. Similarly, the scale in decision becomes very expensive for the *WorstFit* heuristic as all hosts have comparable load. Therefore, we introduced a novel heuristic, called *Utilization-based FirstFit*, to overcome these problems.

3.1 Utilization-based FirstFit

The major idea behind our novel heuristic is to place the load always on non-critical hosts. These hosts are not closed to get overloaded and are also not likely to be released with the next scale in decision. A characteristic example of such a host is a host with a load close to the medium between lower and upper utilization thresholds. The heuristic sorts all hosts with enough capacity based on how critical they are and starts always with the non-critical hosts first. Afterwards, it places the moved operator on the first host of the list.
We use two metrics to determine how critical a host is: its current utilization and the utilization trend. The utilization is categorized into five classes: very low (below the lower utilization threshold), low, medium, high, very high (above the upper utilization threshold). The three classes low, medium, high are derived by equally partitioning the interval between lower and upper threshold into three partitions. Based on the described heuristic, the hosts are sorted using the following class ordering: Medium, Low, High, VeryLow, VeryHigh. If two or more hosts belong to the same class, we sort them based on the recent utilization trend. The utilization trend describes the observed slope of the utilization in the last ten measurements. The slope is determined using linear regression. Hosts with a negative slope (a decreasing utilization) are a preferred target for moved operators.

4 Evaluation

We evaluated the different bin packing heuristics using the hardware provided by the HPI Future SOC lab. Our tests were executed on 10 VM's with 2 cores and 2 GB RAM each. In comparison to the previous Future SOC period, we extended our evaluation from previously five to nine different workloads [7, 9] from the financial, energy domain and Twitter data respectively. All experiments where run with the same utilization thresholds, an upper threshold of 0.8 and a lower threshold of 0.3. Each experiment lasted for 60 minutes. We use two major metrics for our evaluation: the monetary cost and the total number of moved operators. We use a pay per use model according to the Amazon EC2 [2], which charges $\$0.135$ per virtual machine per hour. We scaled the reservation time and the prices to a minimum usage time of one minute due to the short experiment duration. The total number of moved operators are used as an indicator for the effects described previously. A wrong placement decision may lead to many additional operator movements in subsequent scaling decisions. For both metrics a smaller value is preferred.
The achieved performance for different bin packing heuristics is presented in Figure 3. The results show for all workloads a varying difference in the number of moved operators for different methods, e.g. for the workload *Twitter Week1* the method *FirstFit* moved up to 120 operators, while *WorstFit* only moves 38 operators. Overall, our novel method *UtilFirstFit* moves in average the smallest number of operators of all studied heuristics. Also the monetary cost varies based on the used bin packing heuristic, however with a significantly smaller variation. Overall, the smallest monetary cost is measured for the heuristic *BestFit*.
A major effort in this period was spent on fine-tuning

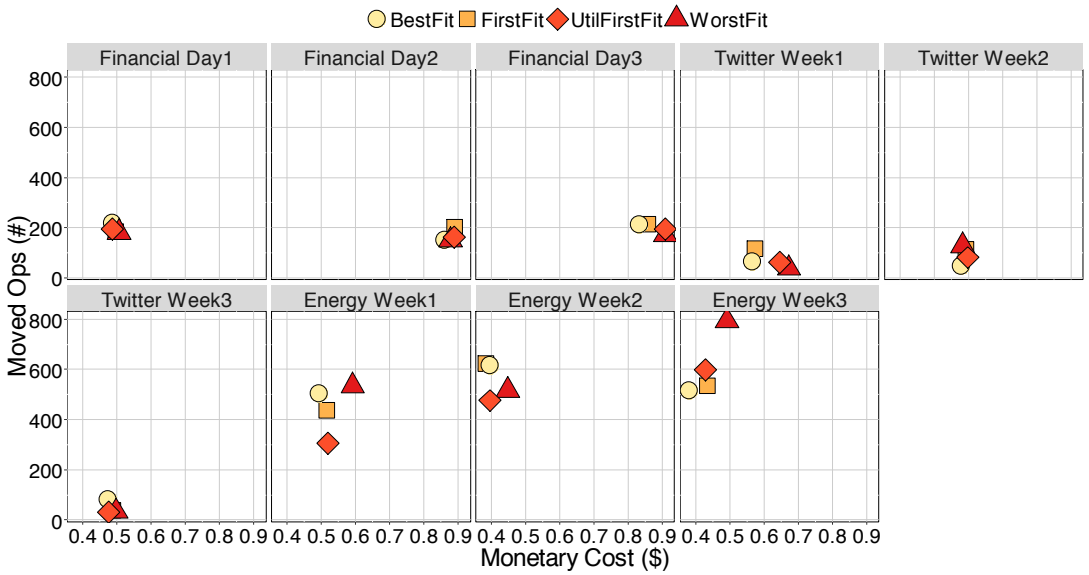


Figure 3: Evaluation Results for Different BinPacking Methods

parameters and testing different configurations. Theses results can not present here in detail due to space constraints. In addition, we started the evaluation of alternative allocation strategies, especially the traffic-aware algorithm of Aniello [3]. We finished the implementation and started the evaluation, but only achieved some preliminary results.

We conclude, that the used allocation strategy influences both the achieved monetary cost and number of moved operators. However, the influence of the bin packing heuristic on both metrics is smaller than for the used thresholds [8, 9] and operator selection strategy [7]. The studied heuristics clearly show a trade-off between these two metrics, where no single best heuristic can be identified. We also saw certain potential to improve the results of well established heuristics by a novel heuristic tailored to our problem.

5 Conclusion

Elastic scalability is an important property of modern data management systems as it is the key to provide a cost efficient execution. This requirement is especially important for data stream management systems, where the workload varies significantly due to changing data stream rates. In context of the HPI Future SoC Lab Spring 2016 we analyzed the elastic scaling data stream management system, where we focused especially on operator placement algorithms. These algorithms decide on which host to move an operator. We studied different well-established heuristics and compared them with a novel heuristic. We saw some potential based on our evaluation results, but the studied heuristics indicate a clear trade-off between the total number of moved operators and the monetary cost.

References

[1] D. J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. Maskey, A. Rasin, E. Ryvkina *et al.*, "The Design of the Borealis Stream Processing Engine," in *CIDR '05: Proceedings of the Second Biennial Conference on Innovative Data Systems Research*, 2005, pp. 277–289.

[2] Amazon, "Amazon EC2," http://aws.amazon.com/ec2/, accessed November 22th, 2015.

[3] L. Aniello, R. Baldoni, and L. Querzoni, "Adaptive online scheduling in storm," in *DEBS '13: Proceedings of the 7th ACM international conference on Distributed event-based systems*. ACM, 2013, pp. 207–218.

[4] R. Barazzutti, T. Heinze, A. Martin, E. Onica, P. Felber, C. Fetzer, Z. Jerzak, M. Pasin, and E. Rivière, "Elastic Scaling of a High-throughput Content-based Publish/Subscribe Engine," in *ICDCS '14: Proceedings of the 2014 34^{th} IEEE International Conference on Distributed Computing Systems*. IEEE, 2014, pp. 567–576.

[5] E. G. Coffman Jr, M. R. Garey, and D. S. Johnson, "Approximation Algorithms for Bin Packing: A Survey," in *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996, pp. 46–93.

[6] H. Ghanbari, B. Simmons, M. Litoiu, and G. Iszlai, "Exploring Alternative Approaches to Implement an Elasticity Policy," in *CLOUD '11:*

Proceedings of the IEEE International Conference on Cloud Computing. IEEE, 2011, pp. 716–723.

[7] T. Heinze, Z. Jerzak, G. Hackenbroich, and C. Fetzer, "Latency-aware Elastic Scaling for Distributed Data Stream Processing Systems," in *DEBS '14: Proceedings of the* 8^{th} *ACM International Conference on Distributed Event-Based Systems*. ACM, 2014, pp. 13–22.

[8] T. Heinze, V. Pappalardo, Z. Jerzak, and C. Fetzer, "Auto-scaling Techniques for Elastic Data Stream Processing," in *ICDEW '14: Workshops Proceedings of the* 30^{th} *International Conference on Data Engineering Workshops*. IEEE, 2014, pp. 296–302.

[9] T. Heinze, L. Roediger, A. Meister, Y. Ji, Z. Jerzak, and C. Fetzer, "Online Parameter Optimization for Elastic Data Stream Processing," in *SoCC '15: Proceedings of the ACM Symposium on Cloud Computing 2015*. ACM, 2015, pp. 276–287.

[10] S. Martello and P. Toth, "Algorithms for Knapsack Problems," *Surveys in Combinatorial Optimization*, vol. 31, pp. 213–258, 1987.

[11] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis," in *SoCC '12: Proceedings of the Third ACM Symposium on Cloud Computing*. ACM, 2012, p. 7.

[12] M. A. Shah, J. M. Hellerstein, S. Chandrasekaran, and M. J. Franklin, "Flux: An Adaptive Partitioning Operator for Continuous Query Systems," in *ICDE '03: Proceedings of the* 19^{th} *IEEE International Conference on Data Engineering*. IEEE, 2003, pp. 25–36.

Multimodal Recurrent Neural Network for Generating Image Captions

Kosala Herath
Singapore University of Technology and Design
kosala.herath.lk@ieee.org

Abstract

Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. In this experiment, I tried to implement a multimodal recurrent neural network for image caption generating and use it for image similarity measurements. In present, there are different approaches to generate descriptions for a given image. Some of them give great results but sometimes they are much complex and not flexible. The image capturing model that I have implemented is very flexible and simple. It uses a recurrent neural network to generate image captions and it has fully described in this document. I trained the model on Flickr8K dataset and it gave good performance. Then I tried to find semantic similarities of images using this image capturing model and got great results. Finally I present future works which can be done by this image capturing model.

1 Introduction

As humans we can summarize and describe most significant facts and their relationships of a complex scene in a few words without thinking twice. When we see an image or a scene first we can identify the objects and their properties in few milliseconds. Then we can identify relationship among these objects and develop descriptions about the full scene with a sentence. This amazing work is done by our brain and neuron system. However, how could we implement this complex model in a computer? In this experiment I have tried to implement this model using computer vision and natural language processing concepts. In this model we would be able to give an image as an input to the model and get well-structured sentence that describe the image content.

Generating a description for an image is very challenging task, however it could be very useful for many fields in the future. It can give great impact on computer vision applications and tasks. As an example it would be give great help for visually impact people to better understanding the world around them. This task is significantly complex than the well-studied image classification or object recognition tasks. In this task the model has to generate description not only about objects in the image, but it also must describe their relationships for each other, their attributes and the activities they are involved in. Also to describe the image to the human it must involve with natural language processing concepts. Therefore there are many field are involved in this image caption generating model.

Most of the computer vision and natural language processing experiments and new attempts have been done separately in the past. They were tried to give solutions for above sub problems. But concept of model with image to description came in last few years. Therefore, this image caption generation model has great value over other image classification and object detection models. In this model which I have done experiments take an image as an input to the system and give a sequence of words which describe the given image in given dictionary. In this experiment I tried to generate sentence with English words.

The main model consists with two sub models which are Deep Convolutional Neural Network (CNN) and Multimodal Recurrent Neural Network (RNN) for sentence generation.

In the last few years object detection and feature recognition was developed rapidly in machine learning field and as result of this CNNs are developed to produce a rich representation of the input image by embedding it to a fixed length vector. This vector represents many features of the image and it can use to so many computer vision tasks. In this model CNN is used to represent the image and

give the features of the image as the output. This feature vector can be used as an input to the multimodal recurrent neural network to generate image description. When we give the features of the image after encoding with CNN, to the RNN it decode the features to a well-organized sentence which describe the image. CNN part and RNN part are used to develop a join model for image caption generation. (See Figure 1)

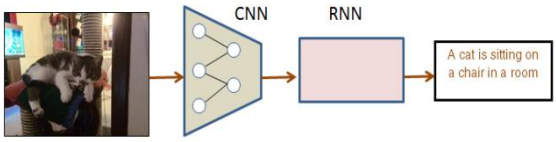


Figure 1: Image Caption Generating Model

There are some models have been implemented for this tasks in past few years and I choose a flexible model that developed by Andrej Karpathy and Li Fei-Fei from University of Stanford [1]. In the paper Karpathy et al. [1] describe an image capturing model with a training data development model for increase performance of the image capturing model. For these experiments I tried only the image capturing model.

After implementing the image capturing model it can use for many different applications in real world problems and tasks. As an example I have tried to measure semantic similarities of two images using this model and It gave great results rather than other methods used in nowadays for image comparing.

2 Model

As shown in Figure 1 the image caption generating model has two main parts. Using both of them we can generate the description for an image. First part is the CNN model which encodes the given image to a feature vector. Then the output of the CNN model is given to the RNN model as an input. Then the RNN model gives a sequence of word that describes the image. To get accurate results we have to optimize these two models.

2.1 Convolutional neural network for feature detection of image

I have done the experiment following Karpathy et al. [1] and in their paper they have not describe about the CNN model that they used for image feature detection in their sentence generating model. Therefore I tried to get the features of images using a VGG Net model that is one of improved versions of the models used by the VGG team in the ILSVRC-2014 competition. These models are developed by following Karen Simonyan et al. [3] paper about very deep convolution networks.

For the experiments I have done, I used 16 layer and 19 layer CNN versions. However these two models gave very similar results for test tasks but the 16 layer version of CNN is fast than 19 layer CNN version. Therefore I used 16 layer CNN version for further experiments.

Let's consider all configurations of convolution neural networks which are described in Karen Simonyan et al. [3] paper. (Table 1)

In the training and testing, the input to this convolution neural network is a fixed size 224 x 224 RGB image. The only preprocessing done in the model is subtracting the mean RGB value, computed on the training set, from each pixel. If we use random image as an input to the model we have to resize image to 224x224 size and subtract the mean RGB value from each pixel. After preprocessing the image it is passed through a stack of convolutional layers, which has filters with a very small receptive field: 3x3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations they also utilize 1 × 1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of convolution layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1 pixel for 3 × 3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2 × 2 pixel window, with stride 2. [3]

A stack of convolutional layers (which has a different depth in different architectures) is followed by three Fully-Connected (FC) layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks. However for this image capturing model we do not want image classification and we only want features of the image which we can give to the RNN model as an input. The multimodal recurrent neural network which developed by following Karpathy et al. [1] is implanted to get image feature input as a 4096 dimension vector

which gives layer before last Fully-connected layer in VGG Net. Therefore I have change the VGG Net implementation of Karen Simonyan et al. [3] to get output as 4096 dimension feature vector. In my implementation of VGG Net I have neglect last fully connected layer and soft-max layer and got the output from above layer of last fully connected layer. All hidden layers are equipped with the rectification (ReLU (Krizhevsky et al., 2012 [2])) non-linearity.
After implementing this Convolution neural network we have to train it over a large amount of images. However I used a pre-trained model that trained on ILSVRC-2012 dataset.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 1: ConvNet configurations of CNNs [3]

In these experiments, I used configuration D (in Table 1) pre-trained convolutional neural network for image feature detection with neglecting last two layers of the configuration. Therefore I got output as a 4096 dimension feature vector. This feature vector can give to the RNN model as an input to generate sentence that describe the image.

2.2 Multimodal recurrent neural network for generating descriptions

In this section describes the multimodal neural network developed by Karpathy et al. [1] for sentence generating for a given image input. The key challenge is the design of a model that can predict a variable-sized sequence of words for a given image. In previously developed language models based on Recurrent Neural Networks (RNNs) [9, 10, 11], this is achieved by defining a probability distribution of the next word in a sequence given the current word and context from previous time steps. In the paper they introduced a simple but effective extension that additionally conditions the generative process on the content of an input image. More formally, during training the Multimodal RNN takes the image pixels I and a sequence of input vectors $(x_1,\ldots, x_T)$. It then computes a sequence of hidden states $(h_1,\ldots, h_T)$ and a sequence of outputs $(y_1,\ldots, y_T)$ by iterating the following recurrence relation for $t = 1$ to T. [1]

$$b_v = W_{hi}[CNN_{\theta_c}(I)]$$
$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b_h + \mathrm{I}(t=1)\odot b_v)$$
$$y_t = soft\max(W_{oh}h_t + b_o)$$

In the equations above, W_{hi}, W_{hx}, W_{hh}, W_{oh}, x_i and b_h, b_o are learnable parameters, and $CNN_{\theta_c}(I)$ is the last layer of a CNN model. x_i is a representation of each word in vocabulary and it can be create using word embedding method. The output vector y_t holds the (unnormalized) log probabilities of words in the dictionary and one additional dimension for a special END token. Note that we provide the image context vector b_v to the RNN only at the first iteration, which Karpathy et al.[1] found to work better than at each time step. They also found that it can help to also pass both b_v, $(W_{hx}x_t)$ through the activation function. A typical size of the hidden layer of the RNN is 512 neurons. (Refer to Figure 2)

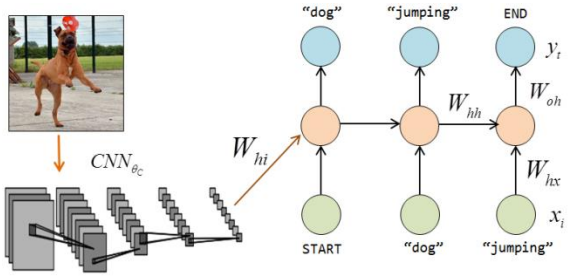


Figure 2: Recurrent neural network

After implementing the above model we can train the model using input data set with images and corresponding image describing sentences. All the training process is descried in Kaparthy et al. [1].
The RNN is trained to combine a word (x_t), the previous context (h_{t-1}) to predict the next word (y_t).
In training method, they condition the RNN's predictions on the image information (b_v) via bias interactions on the first step. The training proceeds as follows (refer to Figure 2): First, set $h_o = \vec{0}$, x_1 to a special START vector, and the desired label y_1 as the first word in the sequence. Analogously, set x_2 to the

word vector of the first word and expect the network to predict the second word, etc. Finally, on the last step when x_T represents the last word, the target label is set to a special END token. The cost function is to maximize the log probability assigned to the target labels (i.e. Softmax classifier). Using cost function and train the model with cost minimization we can reduce the error of the prediction over the time. When we train the model parameters get update and they train to give accurate result. To get more accurate results we have to train the model on larger datasets.

In the test time we can input an image feature vector to the trained RNN model and generate sentence for corresponding image. To predict the sentence, first we have to create image representation b_v, set h_o=0, x_1 to the START vector and compute the distribution over the first word y_1. Then we can sample a word from the distribution (or pick the argmax), set its embedding vector as x_2, and repeat this process until the END token is generated. After generating the END token we can get the well-structured sentence as a output from the RNN model.

3 Experiments

3.1 Datasets

In this experiments I have used Flickr8K [7] and Flickr30K [8] datasets. These datasets contain 8,000 and 31,000 images respectively and each is annotated with 5 sentences using Amazon Mechanical Turk [1]. For Flickr8K I used 5,000 images for training the model and rest are for testing the model after training.

3.2 Implementation

There are two main models in this image caption generator and I have implemented them separately. Firstly I got a pre-trained model of convolutional neural network. It was implemented on MatConvNet which is a toolbox in Matlab for implement neural networks. The VGG Net is a 16 layer very deep convolutional neural network and I change the initial implementation to get the 4096 dimensional vector as the output. I have change the code to neglect the last layer of the fully-connected layers and softmax layer of the CNN model. Then I modify code for preprocessing and now CNN model can get an image which has any resolution as an input.

RNN model has implanted by Kaparthy et al. [1] and I got it for my experiments and make suitable changes to implement this image caption generator. In their model they have implement it using python language with help of scipy and numpy libraries. In this model they have present model training code and image caption prediction code. Firstly I have implement the training model and run it on the SOC lab's tesla server. I have trained the RNN model using Flickr8K dataset and it has to run about two days for get more accurate results. Therefore I had to use much faster server and I have used the SOC lab's tesla server for train the RNN model over two days.

During the process of the training the model I collected some trained models over the time. Then I did tests with each of the trained model and compared their accuracy. All the results are discussed in next section.

After training the model over datasets, I tried to test their accuracy using some test images from the datasets and they gave good results for trained model using Flickr8k dataset. To test the image we only have to input image's feature vector that output from CNN model and get the generate sentence for the image. Example sentences generated by multimodal recurrent network for some test images are shown below in Figure 3. (Other example images with categorization for accuracy are in Appendix A). As shown in the Figure 3 some of the captions have described the image accurately and some of them are got wrong description for given image. However most of the generated descriptions are correct and we can improve the accuracy by training the model with large amount of training data.

3.3 Results and Evaluation

In this section consider about get result for large amount of test images and measuring the accuracy of the trained model for further optimizations.

After training the model using Flickr8K dataset I did experiments with this trained model. During the training process I have saved checkpoint of the training model and then I tried to test each training model checkpoints for get the accuracy of the model.

I test each trained model with Flickr8K test image set which has 1000 test images for testing. I gave all of 1000 image to the model and then got image annotations for each image. Then each generated image description is compared with set of five reference sentence written by humans to calculate BLEU [6] score for each trained model.

a man riding a wave on top of a surfboard

a dog is sitting on a grass field

a little girl in pink shirt is playing with a toy

a group of people standing around a parking meter

a man is jumping in the air with a frisbee

Figure 3: Image caption for example images

All the trained model checkpoints and their BLEU scores are shown in the Table 2. (B-n is BLEU score that uses up to n-grams. High is good in all columns). Each trained model checkpoints are saved in each time period cross validations are done. When the checkpoint number increase that says it has trained much more number of iterations or number of training images. That's mean when we go down through the table the time and number of iterations the model has trained is increase. Therefore, as we can see the BLEU score of the each checkpoint is getting increase when we go down the table. That means if we can give more time and number of iterations to train the model we would get more accurate results from the RNN model. Therefore in future I would like to train this model using MS COCO dataset (with 123,000 images) to improve the model accuracy and performance.

Trained Model	Number of Iterations	BLEU Scores				Time to run on 1000 images
		B-1	B-2	B-3	B-4	
Model_01	3000	23.46	11.36	5.04	2.54	17.51s
Model_02	5000	24.88	12.68	6.34	3.48	16.95s
Model_03	6000	25.42	12.70	6.32	3.44	15.36s
Model_04	10000	26.42	13.78	7.24	4.22	15.57s
Model_05	12000	26.46	13.76	7.36	4.30	15.85s
Model_06	13000	27.66	14.72	8.20	4.92	16.29s

Table 2: BLEU score evaluation for image capturing

3.4 Semantic similarity analysis of images

After training and testing the model I have done some experiments using image caption generator. As an application of the model I tried to analyze the semantic similarities of two images using this model.
In present there are so many methods are used for analyze the image semantic similarity of two images. However some of them are very restricted and has low accuracy. Because analyze the semantic similarities of two image is very difficult and accuracy of the results are very low. However, there are much promising methods to analyze the sentence semantic similarities. Therefore as an answer to the image semantic similarity analysis I tried to use this model.
First, I tried to do some experiments with existing image similarity analyzing method. Most of them use error calculating for each pixel by pixel and give an error value for each two image. If the error value has

zero value then the two images are same. When the error value gets increase that represents each image has not similarities for each other.

I tried to analyze image similarities with pixel by pixel method with Manhattan norm and Zero norm and generate error vale for each two images [4][5]. If the error values are getting low it means two images are have similarities and vice versa. Results for some examples are shown in Appendix B.

Then I tried to analyze image semantic similarities of two images using image capturing model that I implemented. As first step of the process I generated two descriptions for each image and then I compared each generated image captions and compute BLEU score for these two sentences. If the two sentences have semantic similarities the BLEU score gets high value.

Compute a semantic similarity of image images using pixel values of images is very difficult and inflexible method. However to analyze semantic similarities of sentences is much easier than images. Therefore firstly generate image captions and then compare the generate sentences is much promising method for image similarity analysis. This can understand using results which I got in experiments. All the results are shown in Appendix B for comparison the methods.

4 Future Work

Although implemented and tested model's results are encouraging, it's accuracy measurement is much lower than the model that developed by the university of Stanford for image caption generating. Because, I trained my RNN model using only Flickr8K image dataset that has only 8,000 images. Therefore generated sentence are not more similar to the human description for images. To improve the accuracy and the performance of the image capturing model we have to train it on larger dataset like Flickr30K or MS COCO dataset. I would like to train the image capturing model using this larger datasets and get more accurate results for image descriptions. Also if the model can generate more accurate results it will significantly improve image semantic similarity analysis model's performance.

5 Conclusion

After identifying, automatic image description generating is a modern world technology trend then I tried to implement one of good performed image caption generator following Karpathy et al. [1]. I was able to implement the main part of image capturing generator CNN model and RNN model using Matlab and python. Then I trained the model using Flickr8K dataset and measure accuracy after testing on Flickr8K testing dataset. After identifying a good performing trained model I tried to implement image semantic similarity analyzing model using image capturing model and got great results. In the future I would like to improve this model and use to for more real world applications.

6 Reference

[1] A. Karpathy, L. Fei-Fei. Deep visual-semantic Alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014.

[2] A. Krizhevsky, L. Sutskever, G.E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012.

[3] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[4] P. Sinha, R. Russell. A perceptually based comparison of image similarity metrics. *PMID: 22416586*, 2011.

[5] G. Khosla, N. Rajpal, J. Singh. Evaluation of Euclidian and Manhattan metrics in content based image retrieval system. *Jasvinder Singh et al Int. Journal of Engineering Research and Applications, ISSN : 2248-9622, Vol. 4, Issue 9(Version 1), September 2014, pp.43-49*, 2014.

[6] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[7] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 2013.

[8] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014.

[9] T. Mikolov, M. Karafi´at, L. Burget, J. Cernock`y, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

[10] J. L. Elman. Finding structure in time. Cognitive science, 14(2):179–211, 1990.

[11] I. Sutskever, J. Martens, and G. E. Hinton. Generating text with recurrent neural networks. In *ICML*, 2011.

7 Appendixes

7.1 Appendix A

Good Results

A man in a red shirt riding a bike down a dirt path

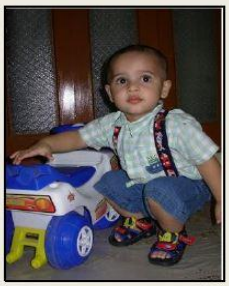
A group of men in red and white uniform are playing soccer

A dog is running through a field

Moderate Results

A young boy in a red shirt is running through a field

A man in a yellow shirt is jumping off a rock into the water



A little girl in a dress is playing with a toy

Unacceptable Results

A group of people are riding a bike in the air

A young boy in a red shirt is jumping into a swimming pool

A young boy in red shirt is standing on a wooden bench with a white dog

7.2 Appendix B

In below results, pixel by pixel comparison is shown in first and then BLEU scores for generated two sentences are shown. Generated descriptions for images are shown below each image. As we can see sentence comparing using image capture generator method gives accurate results rather than pixel by pixel method.

a bird perched on a tree branch in a forest

a bird perched on a tree branch in a forest

Manhattan norm: 0.0 / per pixel: 0.0

Zero norm: 0.0 / per pixel: 0.0

No Error: Perfectly matching

BELU Scores for two sentences:
B-1 :: 100.0
B-2 :: 100.0
B-3 :: 100.0
B-4 :: 100.0
Perfectly Matching

a man is climbing a rock face

a bird perched on a tree branch in a forest

Manhattan norm: 9627893.53988 / per pixel: 106.976594888

Zero norm: 90000 / per pixel: 1.0

Large Error: No similarities

BELU Scores for two sentences:
B-1 :: 28.6
B-2 :: 0.0
B-3 :: 0.0
B-4 :: 0.0
No Similarities

a bird perched on a branch of a tree

a bird perched on a tree branch in a forest

Manhattan norm: 4993540.84138 / per pixel: 55.4837871264

Zero norm: 89902 / per pixel: 0.998911111111

Large Error: No similarities

BELU Scores for two sentences:
B-1 :: 80.0
B-2 :: 66.7
B-3 :: 55.0
B-4 :: 46.7
Matching: There are some similarities in two images

Towards building federations of private clouds using OpenStack

Max Plauth, Felix Eberhard, Frank Feinbube and Andreas Polze
Hasso Plattner Institute for Software Systems Engineering
P.O. Box 90 04 60
14440 Potsdam, Germany
firstname.lastname@hpi.de

Abstract

As a part of our efforts in the Scalable and Secure Infrastructures for Cloud Operations *(SSICLOPS) project, our work during the* Spring 2016 *period was focused on participating in the joined endeavors of building an interconnected federation of OpenStack-based private cloud installations. Providing the foundation for further experiments in the upcoming* Fall 2016 *period, this report aims at providing a conceptual documentation of the status quo.*

1 Introduction

The co-operation among project partners of the *Scalable and Secure Infrastructures for Cloud Operations*[1] (SSICLOPS) project has enabled us to interconnect six OpenStack-based private cloud testbeds located in four different locations (see Figure 1), namely Aalto University (Espoo, Finnland), Helsinki Institute of Physics / CERN (Geneva, Switzerland), NEC Laboratories Europe (Heidelberg, Germany) and the Hasso Plattner Institute for Software Systems Engineering (Potsdam, Germany). With our contribution to the testbed, we are following up on our preceding OpenStack-based projects [3], that aimed at setting up fully virtualized testbeds. The federated testbed was rolled out in order to evaluate developments on inter-cloud transport as well as for testing use-case-specific applications in real-life distributed environments. Further motivations for interconnecting multiple OpenStack installations include enabling resource sharing and resilience, as well as reducing latency between datacenter and clients.

2 Current status

At the beginning of the project, two methods were available for interconnecting multiple OpenStack installations: Running independent systems using public IPs, or using VPN tunnels via the VPNaaS facility

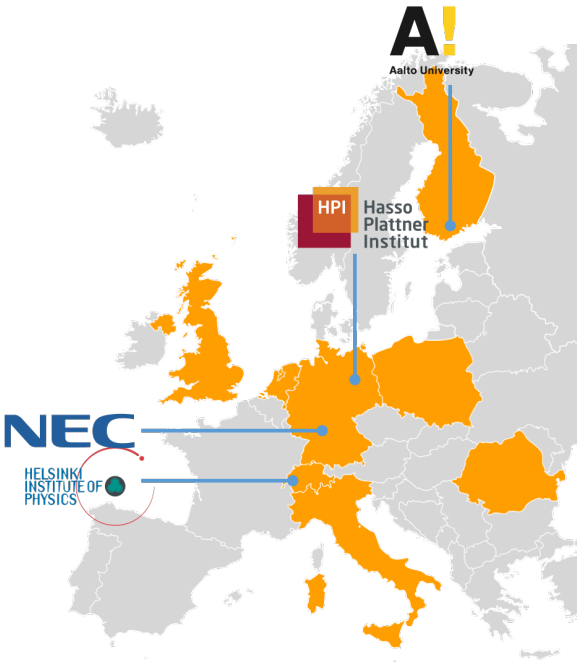


Figure 1: Our federated OpenStack testbed is comprised of independent instances spread across four different geographical locations: Espoo (Finnland), Geneva (Switzerland), Heidelberg (Germany) and Potsdam (Germany).

of OpenStack. The first method usually is not feasible from an administrative point of view, either due to security concerns or the sheer lack of public IPs. Using VPNaaS however, the interconnection is limited to layer 3 connectivity, which inhibits certain use cases, such as using discovery or auto-config mechanisms, which usually rely on multicasts or broadcasts. Furthermore, layer 3 connectivity strongly restricts VM migration between different OpenStack installations.

To address these shortcomings, Maël Kimmerlin from the project partner Aalto University implemented an interconnection agent that facilitates layer 2 connectivity. The interconnection agent also relies on VPN tunnels but allows each OpenStack installation to stretch

[1] https://www.ssiclops.eu

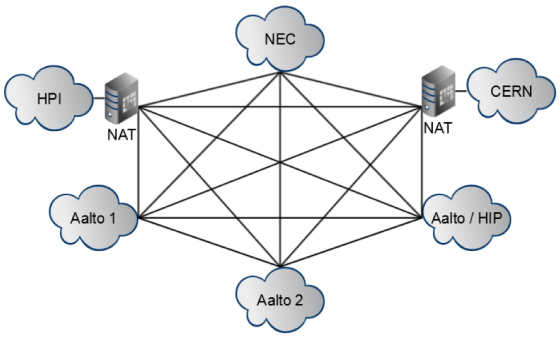


Figure 2: All OpenStack installations are fully interconnected using both the custom layer 2 interconnection agent and VPNaaS for facilitating layer 3 connectivity.

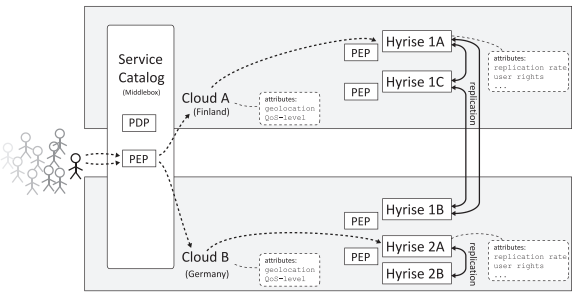


Figure 3: Use case scenario: Users request instances of the *Hyrise-R* in-memory database and annotate their requests with certain policy demands. The *policy decision point* (PDP) acts as the initial entry point and routes requests through a series of *policy enforcement points* (PEP) to process the requests accordingly.

its layer 2 domains across multiple sites. In a stretched network that has been extended to another OpenStack installation, a single broadcast domain is available and the same IP range can be used on all ends. At least from a network-wise point-of-view, VM migration across OpenStack instances should be much easier to implement, even though this has neither been tested nor implemented, yet. Another advantage of the interconnection agent is, that it can take advantage of multiple links between installations to facilitate improved resilience.

We currently operate a federated testbed that is comprised of six OpenStack-based private cloud testbeds operated at Aalto University (Espoo, Finnland), Helsinki Institute of Physics / CERN (Geneva, Switzerland), NEC Laboratories Europe (Heidelberg, Germany) and the Hasso Plattner Institute for Software Systems Engineering (Potsdam, Germany). As depicted in Figure 2, all installations are interconnected via a fully interconnected mesh. In order to quantify the advantages of using the custom layer 2 interconnection agent, we operate an equivalent layer 3 setup using VPNaaS.

3 Outlook

Our goal for the upcoming *Fall 2016* period of the Future SOC Lab is to evaluate an initial proof of concept prototype that implements certain policy language concepts discussed in [1] and [2]. For that purpose, we employ the use case scenario illustrated in Figure 3.

The scenario includes numerous users, where each user requests an instance of the *Hyrise-R* [5] in-memory database in a *Platform as a Service* (PaaS) like manner. However, users impose certain requirements regarding attributes ranging from the coarse-grained properties such as data center location to fine-grained requirements like database configuration parameters. The *policy decision point* (PDP) acts as the main entry point for users requests. With the policy language CPPL [4] developed by our project partners from RWTH Aachen at hands, users can impose requirements on service providers by annotating their requests accordingly.

References

[1] F. Eberhardt, J. Hiller, O. Hohlfeld, S. Klauck, M. Plauth, A. Polze, M. Uflacker, and K. Wehrle. D2.2: Design of inter-cloud security policies, architecture, and annotations for data storage. Technical report, Jan 2016.

[2] F. Eberhardt, M. Plauth, A. Polze, S. Klauck, M. Uflacker, J. Hiller, O. Hohlfeld, and K. Wehrle. D2.1: Report on body of knowledge in secure cloud data storage. Technical report, June 2015.

[3] J. Eschrig, S. Knebel, and N. Kunzmann. Dependable cloud computing with openstack. In D. Bartok, E. van der Walt, J. Lindemann, J. Eschrig, and M. Plauth, editors, *Proceedings of the Third HPI Cloud Symposium Operating the Cloud 2015*, pages 25–32, Potsdam, Germany, 2016. University of Potsdam.

[4] M. Henze, J. Hiller, S. Schmerling, J. H. Ziegeldorf, and K. Wehrle. CPPL: Compact Privacy Policy Language. In *WPES*, 2016. to appear.

[5] J. Lindemann, S. Klauck, and D. Schwalb. A Scalable Query Dispatcher for Hyrise-R. In D. Bartok, E. van der Walt, J. Lindemann, J. Eschrig, and M. Plauth, editors, *Proceedings of the Third HPI Cloud Symposium Operating the Cloud 2015*, pages 25–32, Potsdam, Germany, 2016. University of Potsdam.

Global-Scale Internet Graphs: Vulnerability Analysis & Initial Worm Spread Simulations

Benjamin Fabian
HfT Leipzig & Humboldt Universität zu Berlin
Spandauer Straße 1, 10178 Berlin, Germany
bfabian@wiwi.hu-berlin.de
https://www.researchgate.net/profile/Benjamin_Fabian

Abstract

Based on our integrated traceroute data from global-scale mapping projects to generate comprehensive Internet maps at different abstraction levels, we (a) conducted the main graph analyses with respect to identifying important nodes.

In an evolution of the project, we (b) started to assess several malware strategies that could affect border routers. Their impact will be studied by further simulations on Internet graphs.

1 Introduction

This project [1] aims at developing methods for creating and analyzing a large integrated set of Internet graphs at the IP-interface level as the basis for subsequent examinations. Our analyses include the search for bottlenecks and weak points in the entire Internet topology as well as in the topological connectivity of individual firms and services.

As our project evolved, a novel line of research studies the impact of malware that affects important border routers, and investigates their impact on Internet robustness via graph-based simulations.

2 Research Approach

This project aims at advancing the understanding of the Internet topology by integrating empirical data into a multi-leveled graph model. The main emphasis of our research project is placed on both generating and analyzing a combined global-scale Internet graph at different topological abstraction levels (i.e., IP-interface, Point-of-Presence PoP, Autonomous Systems AS). In this project period, in project line (a) emphasis was placed on the analysis of the generated graphs.

Furthermore, a second line of graph-based research emerged: (b) We simulated the systematic destructive capability of Internet worms that would affect border routers, and began to study their effects.

3 Related Publications

The Institute of Information Systems at Humboldt University Berlin has been conducting research based on graph analysis for several years [1-10].

In particular, robustness analyses and vulnerability assessments of the Internet at the AS-level have been conducted. Large-scale graph analysis has also been applied on the Bitcoin transaction network [3,4] and Twitter use in the political sphere [10].

Further publications based on the current project are under development [11-14]. We list some of them as white papers but note that some of them are not finalized while others are currently under review.

Aspects of our research have also been covered in a major German newspaper, "*Der Tagesspiegel*" [15].

4 Project Plan

Our project requires powerful computation capabilities based on the large-scale memory and multi-core architecture of the HP Converged Cloud and the newly implemented SAP HANA Graph Engine. The project is structured in several phases.

The first phase of the project consisted of data acquisition and pre-processing. The second phase was concerned with extracting the graph at different granularities from the cleansed and combined raw data. The third phase deals with the actual graph analysis of the extracted datasets, which is computationally expensive on such a massive scale. A fourth phase, related to the new project line (b), will study the destructive capabilities of malware spreads via outage simulations.

With the help of the computational power of HPI Future SOC Lab, we are better equipped be able to examine centrality measures, clustering coefficients, shortest paths, and connected components. The fourth

phase consisted (a) of attack and failure simulations of parts of the graphs constructed in phases 1-3.

The novel project line (b) is also carried out on the resources provided by the HPI Future SOC Lab [16].

5 Project Status and Results

5.1 Project Line (a): Quantitative Analyses of the Global Internet Graphs

The advances in this project line are currently under review. Our project advances the understanding of the Internet's structure by pursuing the novel approach of combining data from different large-scale measurement campaigns into a set of integrated Internet graphs at different abstraction levels. Important statistics and graph measures are calculated based on this novel data set, which we will publish to support future research on the Internet topology.

5.2 Project Line (b): Malware Simulations

The project plan in this line of research is described as follows. A vulnerability assessment of the Internet's core components would help to identify critical areas and to build more resilient structures.

The border routers, interconnecting the autonomous systems, constitute a possible bottleneck of the Internet. This work attempts to quantify the impact of attacking these border routers on the autonomous system level connectivity. For this purpose, various worms are simulated. These worms are able to infiltrate the router operating systems of leading suppliers, such as Cisco and Juniper. In order to identify these router operating systems, a TTL-based fingerprinting method is conducted.

The results for the different attack scenarios will be compared and investigated.

5.3 Use of Hardware Resources

The hardware provided by the HPI Future SOC Lab so far included three HP Converged Cloud Blades with 24 x 64-bit CPUs running at a frequency of 1.2 GHz on Ubuntu 14.04. Each of the three machines had 64 GiB memory and was equipped with 1 TiB HDD. This configuration offers an extensive parallelization of tasks.

The calculated results would not have been possible without the support of the HPI. For the intense calculations of the vulnerability analyses, the use of HPI Future SOC Lab resources [16] is crucial. We are very grateful for the continuing support.

6 Conclusion

The phases 1-4 of the project line (a), data integration graph extraction, initial and main graph analysis, have reached important milestones. In future work, we aim to conduct further vulnerability analyses in both research lines (a) and (b).

References

[1] B. Fabian and G. Tilch: Analyzing the Global-Scale Internet Graph at Different Topology Levels: Data Collection and Integration, *HPI Future SOC Lab Day Workshop & Report*, 2015.

[2] A. Baumann and B. Fabian, "How Robust is the Internet? – Insights from Graph Analysis," in *Proceedings of the 9th International Conference on Risks and Security of Internet and Systems (CRiSIS 2014)*, Trento, Italy, Springer, LNCS 8924, 2014.

[3] A. Baumann, B. Fabian, and M. Lischke, "Exploring the Bitcoin Network," in *Proceedings of the 10th International Conference on Web Information Systems and Technologies (WEBIST 2014)*, 2014, pp. 369–374.

[4] M. Lischke, B. Fabian: Analyzing the Bitcoin Network: The First Four Years, *Future Internet* 8(1), March 2016.

[5] B. Fabian, A. Baumann, and J. Lackner, "Topological Analysis of Cloud Service Connectivity," Computers & Industrial Engineering, vol. 88, pp. 151–165, October 2015.

[6] A. Baumann and B. Fabian, "Vulnerability Against Internet Disruptions – A Graph-based Perspective," *Proceedings of the 10th International Conference on Critical Information Infrastructures Security (CRITIS 2015),* Berlin, Germany, October 2015, Springer LNCS 9578.

[7] A. Baumann and B. Fabian, "Who Runs the Internet? Classifying Autonomous Systems into Industries," *Proceedings of the 10th International Conference on Web Information Systems and Technologies (WEBIST),* Barcelona, Spain, April 2014.

[8] A. Baumann and B. Fabian, "Towards Measuring the Geographic and Political Resilience of the Internet," *International Journal of Networking and Virtual Organisations* 12/2013; 13(4):365-384.

[9] M. Huth and B. Fabian: Inferring Business Relationships in the Internet Backbone, *International Journal of Networking and Virtual Organisations*, 2016.

[10] A. Baumann, B. Fabian, S. Lessmann, L. Holzberg: Twitter and the Political Landscape – A Graph Analysis of German Politicians. *Proceedings 24th European Conference on Information Systems* (ECIS 2016), Istanbul, Turkey.

[11] B. Fabian, G. Tilch, A. Baumann, T. Ermakova: Graph Analysis of the Internet TopologyWorking Paper, 2016.

[12] B. Fabian, A. Baumann, M. Ehlert, V. Ververis, T. Ermakova: CORIA – Analyzing Internet Connectivity Risks Using Network Graphs, Working Paper, 2016.

[13] B. Fabian, S. Dombrowski, A. Baumann, T. Ermakova: Cloud Computing Disruptions: Towards Graph-Based Simulations of IP-Level Connectivity, Working Paper, 2016.

[14] B. Fabian, S.Kelkel, A. Baumann, T. Ermakova: Internet Robustness Analysis – Simulation of Worm-Based Router Attacks. Working Paper, 2016.

[15] R. Cisielski: „Schatz, das Internet ist kaputt: Immer mehr Menschen nutzen das Netz, immer mehr Maschinen und Prozesse sind davon abhängig. Doch was passiert, wenn es zur digitalen Apokalypse kommt? In Berlin simulieren Wissenschaftler den Totalausfall". *Der Tagesspiegel,* 24.09.2016, S. 11.

[16] HPI Future SOC Lab. URL: https://hpi.de/forschung/future-soc-lab.html. Accessed 25 Oct 2016.

The Structure of Industrial SAT Instances
Comparing different SAT solvers

Tobias Friedrich, Ralf Rothenberger, and Andrew M. Sutton

Hasso Plattner Institute, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam
{tobias.friedrich,ralf.rothenberger,andrew.sutton}@hpi.de

Abstract

We continue our ongoing project examining non-uniform random distributions of propositional satisfiability formulas. In this phase of the project, we compare the results of different SAT solvers near the phase transition of scale-free propositional satisfiability instances.

1 Introduction

Propositional satisfiability (SAT) is one of the most fundamental problems in computer science. Many practical questions from different domains can be encoded as propositional formula and solved by determining the satisfiability of the resulting formula. A propositional formula is constructed from a set V of n Boolean variables by forming a conjunction

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$$

of m disjunctive clauses where

$$C_i = (\ell_1 \vee \ell_2 \vee \cdots \vee \ell_{k_i}).$$

where $\ell_j \in \{v, \neg v\}$ for some $v \in V$. Here $\neg v$ denotes the logical negation of v. The goal of the decision problem is to decide if there is an assignment to all variables of V so that F evaluates to true. SAT is a central problem in theoretical computer science, but it is also an important practical problem since many difficult combinatorial problems reduce to it.

SAT instances and distributions. A distribution of SAT instances is typically parameterized by n and m and is described by a categorical distribution over all formulas over n variables and m clauses. The most heavily studied distribution of SAT instances is the *uniform* distribution. The uniform distribution is the distribution $U_{n,m}$ of all well-formed CNF formulas on n variables and m clauses where each formula has the same probability of being selected.

Most theoretical work on SAT instances has focused almost exclusively on this uniform distribution $U_{n,m}$. Uniform random formulas are easy to construct, and have shown to be accessible to probabilistic analysis due to their statistical uniformity. Indeed, a long line of successful research has relied on the uniform distribution, and from it, several sophisticated rigorous and non-rigorous techniques have developed for analyzing random structures in general.

Nevertheless, a focus on uniform random instances comes with a risk of driving SAT research in the wrong direction [10] because such instances do not possess the same structural properties as ones encountered in practice. It is well-known that solvers that have been tuned to perform well on one class of instances do not necessarily perform well on another [4], and studying the algorithmics of solvers on uniform random formulas can lead research astray.

The empirical SAT community has expanded their view to study *industrial* instances. Industrial instances arise from problems in practice, such as hardware and software verification, automated planning and scheduling, and circuit design. Empirically, industrial instances appear to have strongly different properties than formulas generated uniformly at random, and as might be expected, SAT solvers behave very differently when applied to them [7, 11].

Furthermore, a number of *non-uniform* random distributions have been recently proposed. These models include regular random [5], geometric [6] and scale-free [1, 2]. The scale-free model is especially promising because the *degree distribution* (distribution of variable occurrence) of instances follows a power-law and this phenomenon has been observed on real-world industrial instances.

Project aim. The goal of this phase of the project was to utilize the parallel computing power of the 1000 node cluster of the Future SOC Lab to (1) generate a

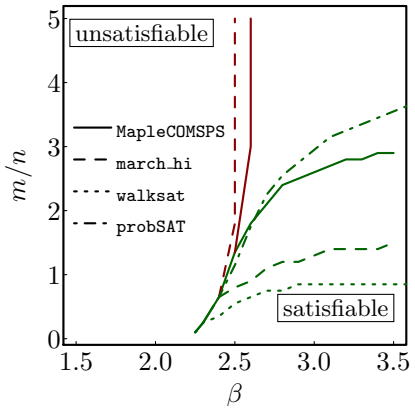


Figure 1: Comparison of threshold bounds proposed by four different solvers for formulas of clause length $k = 3$. As a function of β: the upper bound on density for the unsatisfiable phase is drawn in red; lower bound on density for satisfiable phase drawn in green.

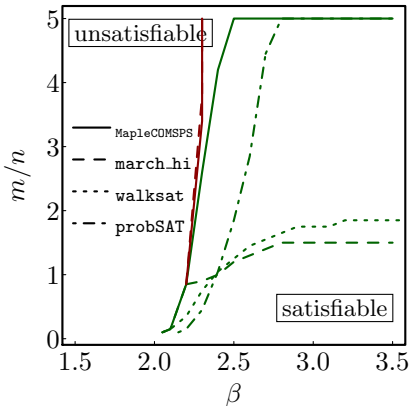


Figure 2: Comparison of threshold bounds proposed by four different solvers for formulas of clause length $k = 4$. As a function of β: the upper bound on density for the unsatisfiable phase is drawn in red; lower bound on density for satisfiable phase drawn in green.

massive set of large random non-uniform (scale-free) formulas and check their satisfiability & hardness and (2) compare a number of SAT solvers in their ability to determine bounds on the satisfiability threshold.

2 Comparing SAT solvers

We compared the performance of four state-of-the-art SAT solvers.

1. `MapleCOMSPS` [12]: a CDCL solver based on `MiniSAT` [8] that implements machine learning in its branching heuristics. Both `MapleCOMSPS` and `MiniSAT` have performed well on industrial benchmarks.

2. `march_hi` [9]: a DPLL-based solver employing look-ahead heuristics to select branching variables.

3. `WalkSAT` [13]: a simple stochastic local search (SLS) solver that is based on a conflict-directed random walk.

4. `probSAT` [3]: a simple probabilistic SLS solver that computes a distribution over variables to flip based on make and break counts.

We used GNU Parallel [14] to distribute a large number of jobs over the cluster. Each job was responsible for generating a set of random scale-free formulas, and then attempting to solve each with each of the solvers listed above within a predetermined time limit. In an interest to eliminate statistical fluctuations that sometimes arise at small problem sizes, we set n very large, specifically $n = 10^6$. For each power-law exponent $\beta = 1.5, 1.6, \ldots, 3.5$ and each m such that $m/n = 1/10, \ldots, 10$ we generated 50 scale-free formulas in the above manner, and ran each solver with a timeout of 15 minutes (900 seconds). If the satisfiability of the formula could not be determined by the solver within this time, the formula is marked as "hard", and its satisfiability state is unknown.

We report the results for formulas of clause length $k = 3$ and $k = 4$ in Figures 1 and 2 respectively. The lines in these plots can be interpreted as follows. At each β value, the highest (resp., lowest) density at which the majority of formulas are successfully determined to be unsatisfiable (resp., satisfiable) yields a proposed upper bound (resp., lower bound) on the threshold at that β. The upper bounds (at the unsat region) are drawn in red and the lower bounds (at the sat region) are drawn in green. Note that the SLS solvers are incomplete, and thus can only propose lower bounds on the threshold.

3 Runtime scaling

We are also interested in how different solvers scale at and below the critical point. To analyze this, we generated many formulas in parallel on the cluster at a fixed density and power law exponent, adjusting n from 10^3 to 10^4 in steps of 100. Fixing $k = 3$, for each value of n, we generate 50 formulas each and run the two complete solvers (`MapleCOMSPS` and `march_hi`). Again we utilize the parallelism of the 1000 node cluster to distribute these jobs.

In Figure 3 we observe in a semi-log plot the scaling of mean solver time as a function of n at the critical point $\beta = 2.6$, $m/n = 2.28$ in the $k = 3$ model (lower left of the "hard" region in Figure 1). At this point (for $n \leq 10^4$), roughly half of the formulas are unsatisfiable. The figure shows the solver times scaling exponentially with similar bases. Reducing the power-law exponent only slightly to $\beta = 2.4$ results in significantly more efficient scaling for both solvers.

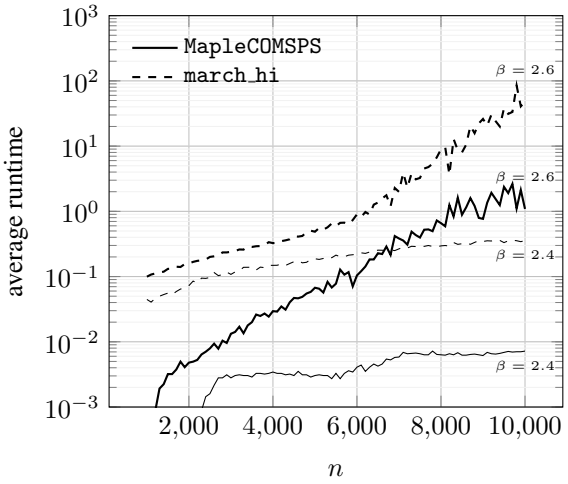


Figure 3: Scaling behavior of `MapleCOMSPS` and `march_hi` at fixed density $m/n = 2.28$ and $k = 3$ at critical point ($\beta = 2.6$) and below critical point ($\beta = 2.4$). Both solvers scale exponentially at the critical point. Slightly below the critical point, both solvers scale more efficiently with problem size.

4 Conclusions

We were able to compare the behavior of a number of different SAT solvers along the phase transition of the power law distribution by executing a massive number of solvers in parallel across the FSOC cluster. We were able to see remarkably similar behavior across two very different styles of backtracking solver in the unsatisfiable phase. We were also surprised that a simple SLS algorithm performs best in the satisfiable phase for $k = 3$.

References

[1] C. Ansótegui, M. L. Bonet, and J. Levy. On the structure of industrial SAT instances. In *15th CP*, pp. 127–141, 2009.

[2] C. Ansótegui, M. L. Bonet, and J. Levy. Towards industrial-like random SAT instances. In *21st IJCAI*, pp. 387–392, 2009.

[3] A. Balint and U. Schöning. probSAT and pprobSAT. In *SAT Competition 2014*, 2014.

[4] M. Birattari. *Tuning Metaheuristics: A Machine Learning Perspective*. Springer, Berlin Heidelberg, 2009.

[5] Y. Boufkhad, O. Dubois, Y. Interian, and B. Selman. Regular random k-SAT: Properties of balanced formulas. In *9th SAT*, pp. 181–200, 2006.

[6] M. Bradonjic and W. Perkins. On sharp thresholds in random geometric graphs. In *18th Intl. Workshop on Randomization and Computation (RANDOM)*, pp. 500–514, 2014.

[7] J. M. Crawford and A. B. Baker. Experimental results on the application of satisfiability algorithms to scheduling problems. In *12th AAAI*, pp. 1092–1097, 1994.

[8] N. Eén and N. Sörensson. An extensible SAT-solver. In *7th SAT*, pp. 502–518, 2004.

[9] M. Heule and H. van Maaren. march_hi. In *SAT Competition 2009*, 2009.

[10] H. Kautz and B. Selman. The state of SAT. *Disc. Appl. Math.*, 155:1514–1524, 2007.

[11] K. Konolige. Easy to be hard: Difficult problems for greedy algorithms. In *4th KR*, pp. 374–378, 1994.

[12] J. H. Liang, C. Oh, V. Ganesh, K. Czarnecki, and P. Poupar. MapleCOMSPS, MapleCOMSPS_lrb, MapleCOMSPS_CHB. In *Proceedings of SAT Competition 2016: Solver and Benchmark Descriptions*, Vol. B-2016-1 of *Department of Computer Science Series of Publications B, University of Helsinki*, pp. 52–53, 2016.

[13] B. Selman, H. A. Kautz, and B. Cohen. Noise strategies for improving local search. In B. Hayes-Roth and R. E. Korf, editors, *Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 - August 4, 1994, Volume 1.*, pp. 337–343. AAAI Press / The MIT Press, 1994.

[14] O. Tange. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36: 42–47, 2011.

Machine Learning Methods for Cognate Production and Semantic Relatedness

Alina Maria Ciobanu
alina.ciobanu@my.fmi.unibuc.ro

Liviu P. Dinu
ldinu@fmi.unibuc.ro

Sergiu Nisioi
sergiu.nisioi@fmi.unibuc.ro

Ana Uban
ana.uban@my.fmi.unibuc.ro

Solomon Marcus Center for Computational Linguistics
Faculty of Mathematics and Computer Science
University of Bucharest

Abstract

In this paper we report the experiments that we ran in the area of cognate production and semantic relatedness, continuing our previous work, and also in other related NLP research problems, using the Future SOC Lab resources. We provide an overview of our project idea, we describe and analyze the results of our experiments, and we discuss possible directions for future work in this area.

1 Introduction

With the growth of linguistic textual data and requirements, natural language processing (NLP) has become more and more resource intensive. In this project we continued our previous work on cognate production and semantic relatedness and we tackled new research problems related to word and language similarity with a focus on the computational resources from HPI Future SOC Lab at our disposal. Our project is divided in two main areas related to word form research, (1) the first area is multilingual and investigates the possibility to extrapolate word belonging to different languages and (2) is focused on the semantic level, to measure the relatedness between words using experimental distributional representations extracted from large corpora.

2 Cognate Production

Cognates are words in different languages having the same etymology and a common ancestor. Given a source language L1, a target language L2 and a word *w* in L1, *cognate production* represents the task of determining the cognate pair of word *w* in L2. In our work, we make use of word pairs and train models at the character level to learn phonologic changes that occur across languages. When working with sequences of characters, it is essential to have powerful computational resources that can handle large dimensional data in order to compile the required models.

2.1 Data

We ran our experiments on two datasets on which previous cognate production results have been reported [2]. Therefore, we were able to evaluate our method in comparison with recent results in this field. We used an English - Spanish (EN - ES) dataset comprising 3,403 cognate pairs and an English - German (EN - DE) dataset comprising 1,002 cognate pairs.

2.2 Our Previous Work

Our first approach to cognate production [6] was based on sequence labeling. Starting from the hypothesis that orthographic changes depend on the context in which they occur, we developed a sequential model system that determines the orthographic form of given words' cognate pairs. Sequence labeling represents the task of assigning a sequence of labels to a sequence of tokens. In our case, the characters of the input word represented the tokens. Our purpose was to obtain, for each input word in the source language, a sequence of labels that concatenated form the input word's cognate pair in the target language. To this end, we employed conditional random fields – CRFs [17] and we ran our experiments using the implementation provided by the Mallet toolkit for machine learning [15].

Alignment. From the alignment of the cognate pairs in the training set we learned orthographic cues and patterns for the changes in spelling, and we inferred the orthographic form of the cognate pairs of the input words from the test set. To align pairs of words we employed the Needleman-Wunsch global alignment

algorithm [16], using a very simple substitution matrix, giving equal scores to all substitutions and disregarding diacritics.

Reranking. The sequential system produced an n-best list of cognates for each input word. We investigated whether the performance of the sequential model can be improved without using additional resources (e.g., a lexicon or a corpus in the target language). We employed a maximum entropy classifier to rerank the n-best output lists provided by the sequential model, using n-grams of characters and word length as features.

Task Setup. We split the data in three subsets for training, development and testing with a ratio of 3:1:1. As features we used n-grams of characters from the input word around the current token, in a window of size w, where $n \in \{1, ..., w\}$. For parameter tuning, we performed a grid search for the number of iterations in $\{1, 5, 10, 25, 50, 100\}$, for the size of the window w in $\{1, 2, 3\}$ and for the order of the CRF in $\{1, 2\}$. We trained the classifier on the training set and evaluated its performance on the development set. For each dataset, we chose the model that obtained the highest instance (word-level) accuracy on the development set and used it to infer cognate pairs for the words in the test set.

Evaluation Measures. To evaluate our system and to compare our results with previous results published on cognate production, we use two evaluation measures:

- Coverage (*COV*): the percentage of input words for which the n-best output list contains the correct cognate pair. We use n = 5.
- Mean reciprocal rank: $MRR(w_i) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{rank_i}$, where *m* is the number of input instances, and $rank_i$ is the position of w_i's cognate pair in the output list.

Results. In Table 1 we report the performance of our sequential system, compared with previous results. For English - Spanish (the larger dataset), our results are comparable to those previously reported, but without using any external resources. The reranking steps improved the results, but not significantly. In order to capture more accurately the context in which orthographic changes between the source and the target language occur, higher-order CRFs – which trigger an exponential increase in the state space – would probably be needed.

2.3 A Deep Learning Approach

Recurrent neural networks have recently been used in numerous NLP studies, offering solid results for different tasks and applications, from language modeling, machine translation, evaluation of machine translation, and semantic measures of similarity [5, 13, 3].

Lang.	Dir.	Prev. (COP)		Exp. #1		Exp. #2	
		COV	MRR	COV	MRR	COV	MRR
EN-ES	→	.65	.54	.59	.45	.62	.45
	←	.68	.48	.63	.51	.67	.52
EN-DE	→	.55	.46	.38	.26	.40	.28
	←	–	–	.40	.31	.41	.32

Table 1: Cognate production results using the sequential system (Exp. #1 – sequential model, Exp. #2 – sequential model with reranking) compared to previous results (COP – [2]).

Our deep learning approach uses sequence-to-sequence models to learn the phonological changes that occur between cognate pairs across languages. These models, proved to be effective on machine translation, recently deep learning-based models have obtained the best machine translation scores in the shared tasks organized by the Workshops and Conferences of Machine Translation [4]. While the results with these models worked outstandingly well on other tasks, on our particular dataset, deep learning approaches managed to get results comparable or even lower than the ones obtained using simpler CRF-based methods. We believe there are two main reasons behind this, (1) the dataset as described in the previous sections is not sufficiently large to train models that are deep enough and (2) additional research is needed in order to make the models robust to small amounts of data. It is yet an open problem how to define *small* when working with different types of data and representations.

3 Semantic Relatedness

In a separate experiment, we studied the problem of word-to-word semantic relatedness, by experimenting with some alternative semantic representations of words, in a distributional semantics framework. We evaluated the performance of these representations for measuring semantic distance between words, as well as how well they can capture semantic similarity as opposed to mere semantic relatedness.

3.1 Our Previous Work

We based our approach on some previous work [7] where we proposed a ranking-based representation of word meaning, using the contexts of words in a corpus. In these experiments, we represented target words as rankings of all co-occurring words in a text corpus, and used distance metrics between rankings (such as Jaro [12] or Rank distance [9]) to compute semantic similarity scores between pairs of words, to be compared to a gold standard (*WS-353 Test*).

3.2 Data and Methodology

We used the publicly available *Wacky* [1] corpus to compute the co-occurrence frequencies. As an addition to our previous work, we extended the approach by also considering the synonyms of the target words, and computed the similarity score between two words as a function (the average or the maximum) of the distances between all possible pairs in the Cartesian product of the two words' synonym sets.
We tested the method on two different gold standards, *WS-353 Test* [10] and *Simlex-999* [11], by comparing the relatedness score given by our algorithm with the scores given by humans for the word pairs in the gold standards. While *WS-353* contains word pairs and a score indicating their semantic relatedness, *Simlex-999* focuses more on semantic similarity.

3.3 Results

The new approach obtained better correlation with the gold standard on *Simlex-999*, but on *WS-353 Test* results were not improved compared to the old method. Nevertheless, the better results on *Simlex-999* suggest our new approach may bring an improvement in regards to capturing semantic similarity, and thus distinguishing between the word pairs that are very close in meaning and those that are more loosely related.

4 Related Research Problems

Besides the main subjects of our project – cognate production and semantic relatedness – there are several other related NLP research problems, that we are interested in.

4.1 Syntactic Similarity

The syntactic similarity measures the relatedness between the language-specific syntactic properties. We are interested in developing a computational method for determining the syntactic similarity between languages. We are investigating multiple approaches and metrics, running a large-scale experiment on 17 languages belonging to various language families.

4.2 Discriminating between Similar Languages

Automatic language identification is the task of determining the language in which a piece of text is written using computational methods. Although language identification has been intensively studied in the recent period, it is still a challenging research problem for very similar languages and language varieties. We participated in the DSL 2016 shared task [14], which tackled two interesting aspects of language identification: similar language and language varieties (with in-domain and out-of-domain – social media data – test sets) and Arabic dialects. We submitted our results [8] in the closed track of sub-task 1 (Similar languages and language varieties) and sub-task 2 (Arabic dialects). For sub-task 1 we used a logistic regression classifier with tf-idf feature weighting and for sub-task 2 a character-based string kernel with an SVM classifier. Our approach worked surprisingly well for out-of-domain, social media data, with 0.898 accuracy (3rd place) for dataset B1 and 0.838 accuracy (4th place) for dataset B2.

5 Future SOC Lab Resources

We have requested from the Lab several resources which we have used in our experiments: a multi-core RX600S5-2 server with 4 x Xeon (Nehalem EX) X7550 CPUs, 1024 GB RAM, 4 x 146 GB HDDs. This server has been used for the majority of experiments that only require CPU power, such as CRF, similarity measures and kernel computation. For the deep learning methods employed, we used the FluiDyna server with 2 Xeon (Nehalem) E5620 CPUs, 24 GB RAM and co-processors: 2 NVIDIA Tesla K20X (6GB GDDR5) and 2 Intel Xeon Phi 5110p (8GB). The power provided by GPUs is essential when working with deep learning since it can compute fast matrix operations and computation of gradients. Having these resources at our disposal was essential to research and compare a range of models and parameters that we employed to accomplish our tasks.

6 Conclusions and Future Work

Our research was not necessarily focused and directed in order to obtain the best possible results, rather to explore *uncharted territories* related to semantic similarity and cognate production. While our results on cognate production are comparable to previous research, the deep learning sequence to sequence model we have employed proves to require additional amounts of data in order to be able to learn rules that generalize well. For this, we plan to work on building extended multilingual dictionaries of cognates in a semi-automatic way and explore the possibility to apply reinforcement learning strategies to cover the missing cases to better model language. For semantic similarity, our main goal is to further investigate the reasons behind the differences on *Simlex-999* compared to *WS-353*, which includes further feature analysis and tuning. Last but not least, creating specialized deep learning for small data is yet an open research problem which requires not only powerful computational resources, but also advanced research into neural networks and learning strategies.

References

[1] M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226, 2009.

[2] L. Beinborn, T. Zesch, and I. Gurevych. Cognate Production using Character-based Machine Translation. In *Proceedings of the 6th International Joint Conference on Natural Language Processing, IJCNLP 2013*, pages 883–891, 2013.

[3] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[4] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. Jimeno Yepes, P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Neveol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics.

[5] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734, 2014.

[6] A. M. Ciobanu. Sequence Labeling for Cognate Production. In *Procedia Computer Science Volume 96, Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 20th International Conference KES 2016*, pages 1391–1399, 2016.

[7] A. M. Ciobanu and A. Dinu. Alternative Measures of Word Relatedness in Distributional Semantics. In *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*, pages 80–84, 2013.

[8] A. M. Ciobanu, S. Nisioi, and L. Dinu. Vanilla Classifiers for Distinguishing between Similar Languages. In *Proceedings of the 3rd Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (VarDial)*, 2016.

[9] L. P. Dinu. Rank distance with applications in similarity of natural languages. *Fundamenta Informaticae*, 64(1-4):135–149, 2005.

[10] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.

[11] F. Hill, R. Reichart, and A. Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 2016.

[12] M. A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.

[13] A. Karpathy, J. Johnson, and F. Li. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078, 2015.

[14] S. Malmasi, M. Zampieri, N. Ljubešić, P. Nakov, A. Ali, and J. Tiedemann. Discriminating between Similar Languages and Arabic Dialect Identification: A Report on the Third DSL Shared Task. In *Proceedings of the 3rd Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (VarDial)*, Osaka, Japan, 2016.

[15] A. K. McCallum. MALLET: A Machine Learning for Language Toolkit. 2002.

[16] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[17] C. A. Sutton and A. McCallum. An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.