DATA ANALYTICS 2013

The Second International Conference on Data Analytics

FEOSW 2013

The Second International Workshop on Finance and Economics on the Semantic Web

ISBN: 978-1-61208-295-0

September 29 - October 3, 2013

Porto, Portugal

DATA ANALYTICS 2013 Editors

Friedrich Laux, Reutlingen University, Germany

DATA ANALYTICS 2013

Foreword

The Second International Conference on Data Analytics (DATA ANALYTICS 2013), held between September 29 and October 3, 2013 in Porto, Portugal, continued a series of events on fundamentals in supporting data analytics, special mechanisms and features of applying principles of data analytics, application-oriented analytics, and target-area analytics.

Processing of terabytes to petabytes of data, or incorporating non-structural data and multi-structured data sources and types require advanced analytics and data science mechanisms for both raw and partially-processed information. Despite considerable advancements on high performance, large storage, and high computation power, there are challenges in identifying, clustering, classifying, and interpreting of a large spectrum of information.

We take here the opportunity to warmly thank all the members of the DATA ANALYTICS 2013 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to DATA ANALYTICS 2013. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the DATA ANALYTICS 2013 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that DATA ANALYTICS 2013 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of data analytics.

We are convinced that the participants found the event useful and communications very open. We hope that Porto, Portugal, provided a pleasant environment during the conference and everyone saved some time to enjoy the charm of the city.

DATA ANALYTICS 2013 Chairs:

DATA ANALYTICS General Chair

Sandjai Bhulai, VU University Amsterdam, The Netherlands

DATA ANALYTICS 2013

Committee

DATA ANALYTICS General Chair

Sandjai Bhulai, VU University Amsterdam, The Netherlands

DATA ANALYTICS 2013 Technical Program Committee

Mohd Helmy Abd Wahab, Universiti Tun Hussein Onn Malaysia, Malaysia
Sayed Abdel-Wahab, Sadat Academy for Management Sciences, Egypt
Rajeev Agrawal, North Carolina A&T State University - Greensboro, USA
Fabricio Alves Barbosa da Silva, Brazilian Army Technological Center - Rio de Janeiro, Brazil
Fabrizio Angiulli, University of Calabria, Italy
Annalisa Appice, Università degli Studi di Bari Aldo Moro, Italy
Giuliano Armano, University of Cagliari, Italy
Ryan G. Benton, University of Louisiana at Lafayette, USA
Erik Buchmann, Karlsruhe Institute of Technology, Germany
Luca Cagliero, Politecnico di Torino, Italy
Huiping Cao, New Mexico State University, USA
Michelangelo Ceci, University of Bari, Italy
Federica Cena, Università degli Studi di Torino, Italy
Lijun Chang, University of New South Wales, Australia
Qiming Chen, HP Labs - Palo Alto, USA
Been-Chian Chien, National University of Tainan, Taiwan
Fabio Crestani, University of Lugano (USI), Switzerland
Alain Crolotte, Teradata Corporation - El Segundo, USA
Bo Dai, Purdue University, U.S.A.
Tran Khanh Dang, National University of Ho Chi Minh City, Vietnam
Jérôme Darmont, Université de Lyon - Bron, France
Ernesto William De Luca, University of Applied Sciences Potsdam, Gernmany
Kamil Dimililer, Near East University, Cyprus
Shifei Ding, China University of Mining and Technology - Xuzhou City, China
Sherif Elfayoumy, University of North Florida, USA
Yi Fang, Purdue University - West Lafayette, USA
Wai-keung Fung, University of Manitoba, Canada
Matjaz Gams, Jozef Stefan Institute - Ljubljana, Slovenia
Paolo Garza, Dipartimento di Automatica e Informatica Politecnico di Torino, Italy
Shlomo Geva, Queensland University of Technology - Brisbane, Australia
Ahmad Ghazal, Teradata Corporation - El Segundo, USA
Amer Goneid, American University in Cairo, Egypt
Raju Gottumukkala, University of Louisiana at Lafayette, USA
William Grosky, University of Michigan - Dearborn, USA
Tudor Groza, The University of Queensland, Australia

Jerzy W. Grzymala-Busse, University of Kansas - Lawrence, USA
Shengbo Guo, Xerox Research Centre Europe, France
Tiziana Guzzo, National Research Council/Institute of Research on Population and Social Policies - Rome, Italy
Michael Hahsler, Southern Methodist University, U.S.A.
Sven Hartmann, TU-Clausthal, Germany
Quang Hoang, Hue University, Vietnam
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Yi Hu, Northern Kentucky University - Highland Heights, USA
Jun (Luke) Huan, University of Kansas - Lawrence, USA
Mao Lin Huang, University of Technology - Sydney, Australia
Stratos Idreos, Centrum Wiskunde & Informatica (CWI), Netherlands
Sergio Ilarri, University of Zaragoza, Spain
Ali Jarvandi, George Washington University, U.S.A.
Farhana Kabir, Intel, U.S.A.
Prabhanjan Kambadur, IBM TJ Watson Research Center, USA
Daniel Kimmig, Karlsruhe Institute of Technology (KIT), Germany
Boris Kovalerchuk, Central Washington University, U.S.A.
Michal Kratky, VŠB-Technical University of Ostrava, Czech Republic
Dominique Laurent, University of Cergy Pontoise, France
Kerstin Lemke-Rust, Hochschule Bonn-Rhein-Sieg, Germany
Johannes Leveling, Dublin City University, Ireland
Tao Li, Florida International University, USA
Dan Lin, Missouri University of Science and Technology Rolla, U.S.A.
Wen-Yang Lin, National University of Kaohsiung, Taiwan
Weimo Liu, Fudan University, China
Xumin Liu, Rochester Institute of Technology, USA
Corrado Loglisci, University of Bari, Italy
Yi Lu, Prairie View A&M University, USA
Shuai Ma, Beihang University, China
Prabhat Mahanti, University of New Brunswick, Canada
Serge Mankovski, CA Technologies, Spain
Yannis Manolopoulos, Aristotle University of Thessaloniki, Greece
Archil Maysuradze, Lomonosov Moscow State University, Russia
Michele Melchiori, Università degli Studi di Brescia, Italy
Shicong Meng, Georgia Institute of Technology, USA
George Michailidis, University of Michigan, USA
Victor Muntés Mulero, CA Technologies, Spain
Sumit Negi, IBM Research, India
Sadegh Nobari, Sharif University of Technology, Iran
Panos M. Pardalos, University of Florida, USA
Dhaval Patel, Indian Institute of Technology-Roorkee, India
Jan Platoš, VSB-Technical University of Ostrava, Czech Republic
Ivan Radev, South Carolina State University, USA
Zbigniew W. Ras, University of North Carolina - Charlotte, USA & Warsaw University of Technology, Poland
Jan Rauch, University of Economics - Prague, Czech Republic
Manjeet Rege, Rochester Institute of Technology, USA

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

DNA: An Online Algorithm for Credit Card Fraud Detection for Games Merchants

Michael Schaidnagel
D-72072 Tübingen, Germany
Michael.Schaidnagel@web.de

Ilia Petrov, Fritz Laux
Data Management Lab
Reutlingen University
D-72762 Reutlingen, Germany
{Ilia.Petrov | Fritz.Laux}@reutlingen-university.de

***Abstract*—Online credit card fraud represents a significant challenge to online merchants. In 2011 alone, the total loss due to credit card fraud amounted to $ 7.60 billion with a clear upward trend. Especially online games merchants have difficulties applying standard fraud detection algorithms to achieve timely and accurate detection. The present paper introduces a novel approach for online fraud detection, called DNA. It is based on a formula which uses attributes that are derived from a sequence of transactions. The influence of these attributes on the result of the formula reveals additional information about this sequence. The result represents a fraud level indicator, serving as a classification threshold. A systematic approach for finding these attributes and the mode of operation of the algorithm is given in detail. The experimental evaluation against several standard algorithms on a real life data set demonstrates the superior fraud detection performance of the DNA approach (16.25 % better fraud detection accuracy, 99.59 % precision and low response time). In addition to that, several experiments were conducted in order to show the good scalability of the suggested algorithm.**

Keywords- binary classification, credit card fraud, online environment

I. INTRODUCTION

The approximate global business volume of the computer gaming industry in total rose from $ 20 billion in 2001 to $ 65 billion in 2011 [1]. It is estimated to grow by 10.6 % in 2013. New technology developments, such as browser games and Massive Multiplayer Online Games have created new business models (based on micropayments) for online games merchants. Both, technology and business model affect the customers payment behavior. Their first choice for performing online payments is the credit card. The downside of this development is an increase in online credit card fraud, which continues to pose a big threat for online merchants. The total loss due to credit card fraud rose to $ 7.60 billion in 2011 [2] and is supposed to increase further. Especially online games merchants have difficulties applying standard techniques for fraud detection. The reason for this is the lack of personal information about their customers (e.g., real names and postal address for address verification) as well as the need for real time classification.

Problem definition: There are a number of constraints, which make it difficult to apply the traditional algorithms for credit card fraud detection. Players do not feel comfortable to reveal their real names and addresses in an online gaming environment. This lack of financial data in addition to the short transaction histories of players makes it difficult to apply standard techniques. Furthermore, the real time nature of business makes it necessary to be able to apply an algorithm in real time, or near real-time, in order to reject fraudulent transactions at authorization time. Most of the techniques proposed so far are bulk oriented and designed for offline batch processing.

Contributions: We present a novel algorithm, which is able to handle the scarce data situation by deriving attributes out of a sequence of transactions. These attributes are normalized, weighted and arranged in a way that enables a simple formula to recognize different fraud behavior patterns. In order to assess transactions without any history, a concept of cultural clusters was introduced to help classifying those transactions. In addition to that, a metric for assessing the suitability of attributes, their influence on the fraud level as well as the calculation of the threshold are introduced. The DNA approach performs 16.26 % better than the best standard method (Bayesian Net) and achieves an almost perfect 99.59 % precision. In addition, the DNA approach scales better than other approaches with increasing data volumes, while offering acceptable response/detection times.

The presented paper is structured as follows: Section II will give an overview of the related work, which describes different data mining algorithms normally suggested for this problem. They are also part of the experimental evaluation. Section III will detail our suggested method and describe the major components. The suggested method is applied to a real life data set in Section IV. Section V concludes the results and mentions a few points for future work.

II. RELATED WORK

So far, there have been many data mining algorithms applied in order to detect credit card fraud [3]. Please note that we do not go into details here on how they work. All mentioned methods have been implemented and will be compared in terms of fraud detection performance in Section IV.

Artificial Neural Network (ANN): Gosh and Reilly [4] were the first ones to adapt Neural Networks on credit card fraud detection. Other authors such as Dorronsoro et al. [5], Brause et al. [6] and Maes et al. [7] have also implemented ANNs in real life applications. ANNs in general are too dependent on meaningful attributes, which should not

 ISBN: 978-1-61208-295-0

necessarily be available. The information gain from such attributes is too low to be utilized in ANNs.

Bayesian Belief Network (BBN): The first implementation for fraud detection was done by Ezawa et al. [8]. Other recent implementations are Lam et al. [9], Maes et al. [7] and Gadi et al. [10]. However, some data set do not provide enough attributes in order to construct a suitable network.

Hidden Markov Model (HMM): In recent years several research groups applied this model for fraud detection. Srivastava et al. [11] have conducted a very systematic and thorough research in their work. Other implementations were done by Mhamane et al. [12], Bhusari et al. [13] and Dhok [14]. A classic and comprehensive introduction to the topic of HMM was published by Rabbiner and Juang [15] and also Stamp [16] is worth reading for introductory purposes. HMMs in general are only able to utilize a single numeric attribute for their prediction, which is insufficient for a proper classification.

Decision Tree (DT): The biggest impact on how Decision Trees are built had Quinlan [17] in the late 90s. There have been some applications on fraud detection in recent years, e.g., Minegishi et al. [18]. Other mentionable fraud detection implementations are Sahin and Duman [19], Sherly et al. [20] and Gadi et al. [21]. DTs in general suffer the same insufficiencies as ANNs.

III. PROPOSED METHOD

The algorithm is named after the famous Deoxyribonucleic Acid (DNA), which is the basis of all living organisms. The DNA is able to store very complex information with just four basic components (the so-called nucleotides). The crucial insight here is that not only the sequence of these nucleotides is important, but also their interconnection. Similar to nature, we derive attributes out of the sequence of transactions and then take a look at their influence to our labels (fraud, genuine). The remainder of this section will now detail the different parts necessary for building the DNA approach.

A. Cultural Clusters

The Cultural Clusters were introduced in order to help classifying transactions without any history. The idea behind it was to get as much information out of the given attributes as possible. These attributes include the origin of the user (IP country) and the origin of the credit card used in a transaction (BIN country – BIN is an abbreviation for Bank Identification Number: first 6 digits of a credit card number, enables to locate the card issuing bank of the cardholder). Cultural related countries form clusters; which are roughly based on continents. A range of weights is assigned to each cluster. Every country is assigned with a specific weight within its cluster´s range depending on its cultural distance to its cluster center and the risk of the county of being defrauded. The weight of a country within a certain cultural cluster is set empirically and can be subject for adaption, in case the fraudulent behavior changes. In other words: the weight of a country lies within the range of its cultural cluster and is set by an initial value, based on the experience of a fraud expert. If cards from this country turn out to be defrauded frequently, the weight can be increased (within the limits of its cultural clusters). This will increase the risk value of a country pair, which can be calculated as it can be seen in equation (1):

$$risk = |weight\ (IP\ country) - weight(BIN\ country)| \quad (1)$$

This value will be low for country pairs within the own country cluster (e.g., a user from Sweden tries to use a card originated in Norway) or 0 if the user and the corresponding card are from the same country. On the other hand, this value increases if there is a suspicious country pair involved (cross-cultural cluster). This simple metric allows depicting complex risk relationships between several countries.

B. Building sequence based attributes

As briefly noted at the beginning of this section, the DNA approach does not only rely on the risk assessment of the involved country pairs of a transaction. Furthermore, multiple other sequential based attributes are used to enhance the fraud detection performance. The basic process starts by selecting an attribute which is used to identify associated transactions. An example for such an attribute could be the account number or email address of a user. In this work we use the term sequence to refer to all transactions belonging to a certain user email address. In our case, the attribute email address represents the sequencing attribute which is used to build grouped data entries. Please note that the transactions of a sequence are sorted by the transactions timestamp prior to aggregation. An excerpt of the grouped, intermediate data is shown in Figure 1.

timestamp	userEmail	transaction no	a_1	a_2
01.01.2012	userEmail2	t1	value_X	value_A
03.01.2012	userEmail2	t2	value_Y	value_B
18.01.2012	userEmail2	t3	value_Z	value_A

aggreg. Attrib. a_1	aggreg. Attrib. a_2	sequence identifier	label
1	10	userEmail1	fraud
3	2	userEmail2	genuine
4	9	userEmail3	genuine
5	10	userEmail4	genuine
2	4	userEmail5	genuine

used for parameter calculation

Figure 1. Schematic representation for the sequenciation (grouping) process

During aggregation, we used typical operators, such as sum or count of distinct values and calculated the following parameters of the aggregated attributes for both of our labels (fraud, genuine) and overall for all sequences (see Table I):

TABLE I. AGGREGATED PARAMETERS FOR EXAMPLE ATTRIBUTE *SUMCREDITCARDTOKEN*

parameter name	genuine	fraudulent	total
Maximum value	29	38	38
Minimum value	0	0	0
Average	1.19	2.61	1.34
Standard deviation	0.62	2.85	1.07

Based on these values, we are able to estimate whether an attribute is suitable for the DNA approach or not. As it can

be seen the parameter minimum alone, for example, is not able to sufficiently segregate both labels (the average of both labels yield more segregation). Adequate attributes have a great distance between the average fraud and average genuine value. The corresponding standard deviation boundaries should, if possible, not overlap. Figure 2 shows this based on the example attribute *sumCreditCardToken*. The attribute was part of the data set described in Section IV. It was created by counting up all distinct credit cards, which were used by a certain user_email. For the given example case, we calculated the values for the aggregation parameters (as depicted in Table 1 column two to four).

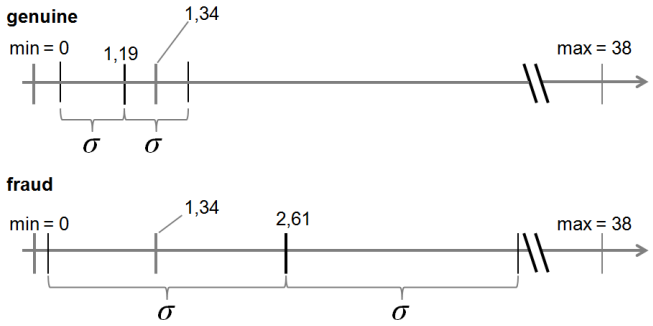


Figure 2. Example distribution for used credit cards per user for two classes of labels (fraud, genuine)

As it can be seen in Figure 2, the depicted attribute is not perfect, since the boundaries of the standard deviation of fraudulent transactions overlap with the area of standard deviation of genuine transactions. However, the distance between the average fraud value and average genuine value is suitable for classification. We can also see that the average fraud value is on the right hand side of the total average value and therefore tends to larger values in a fraudulent case. This indicates that the attribute *sumCreditCardToken* should be placed in the numerator of (4), since it will tend to its maximum if a sequence is fraudulent.

For our work, we were able to identify six suitable attributes:

- *sumCreditCard:* # of distinct credit cards per user
- *sumTransactionStats:* # of transaction status 'rejected'
- *sumSuccessfullTrans* # of completed transactions
- *avgDensity:* average distance between dates
- *sumCountries:* # of distinct countries involved
- *sumDates:* # of distinct dates involved

C. Formula and Weighting

The above-described process is repeated for each of the n available attributes in the given data set [see also (2)] apart from the aggregation attribute and the label attribute. The training data set T consists of transactions t_i which are described by attributes $a_i \in A$, see also Formula (2). Thereby A consists of all original attributes of a transaction a_i and T denotes the set of available transactions with attributes from A. These transactions are grouped into sequences S, which consist of various combined attributes b_i, see also (3). These combinations can either be done by functions $b_i \in f(t_i(a_j))$ or by $b_i \in f(a_i, a_j)$.

The attributes are normalized with the min-max normalization [3, p. 114] to bring the different attributes on the same numerical level (ranging from 0 to 1). We distinguish between two types of attributes, as briefly described above. These attributes are referred as $b_i \in F$, where F is the set of derived attributes that tend to 1 if normalized and are summed up in the nominator of (4). The denominator, in contrary, is composed of a second type of attributes $b_i \in E$, which tend to 0 if normalized with min-max normalization. If the quotient of the normalization expression is not defined, it will be discarded.

$$T = \{t_i(a_1, a_2, \dots, a_n) | \forall i = 1, \dots, n: \quad a_i \in A\} \quad (2)$$

$$S = \left\{ s(b_1, b_2, \dots, b_m) \middle| \begin{matrix} \forall k \in 1,\dots,m: b_k \in f(a_i, a_j) & i,j \in 1,\dots,n \\ \forall k \in 1,\dots,m: b_k \in f(t_i(a_j)) & i \in 1,\dots,|T| \\ & j \in 1,\dots,n \end{matrix} \right\} \quad (3)$$

$$riskLevel = \frac{\sum_{b_i \in F} \left(\frac{b_i - \min_S b_i}{\max_S b_i - \min_S b_i} \right) * \alpha_{b_i}}{\sum_{b_i \in E} \left(\frac{b_i - \min_S b_i}{\max_S b_i - \min_S b_i} \right) * \beta_{b_i}} \quad (4)$$

The next step is to weight the normalized attributes according to their importance for fraud detection. We use parameter $\alpha_{b_i} > 0$ for F attributes and $\beta_{b_i} > 0$ for E attributes to achieve this task. Two sets of parameters are necessary since attributes used for the denominator need to be scaled down, in order to increase their significance. The attributes in the nominator are increased in significance, if the α_{b_i} is scaled up. The weight parameters α_{b_i} and β_{b_i} are determined empirically.

D. Threshold Selection

Formula (4) is an indicator on how prevalent fraudulent attributes are for a particular transaction. The last step for applying the DNA approach for fraud detection is to determine a threshold value whose violation will lead to the classification "fraudulent transaction". As mentioned above, this threshold is determined empirically by undertaking a series of tests with a set of thresholds (e.g., from 0 to 100). Accuracy metrics such as Precision P, Recall R and score F1 (5) are then determined for each assumed threshold. Thereby F1 represents the harmonic mean of Precision and Recall and is used to rank the performance of different methods in the experimental evaluation:

$$F1 = 2 * \frac{P * R}{P + T} \quad (5)$$

The development of these performance measures over different threshold values is depicted Figure 3 for an example case:

 ISBN: 978-1-61208-295-0

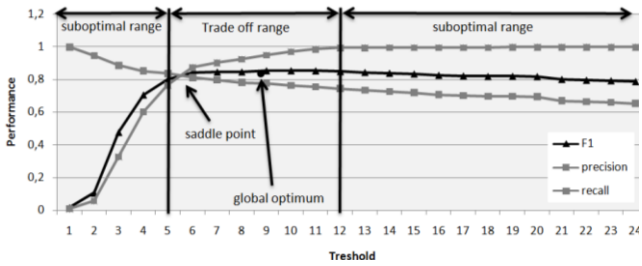


Figure 3. Determining threshold value for DNA approach

Accuracy indicators (Figure 3) are increasing fast until threshold value 5. It is not reasonable to select a threshold lower than 5, since the F1 is far from the global optimum. From threshold 5 on, there is an intersection point, which will keep the F1 near the global optimum. This second range is called "trade-off range" and spans up to threshold value 12, in the case depicted in Figure 3. Within this range the merchant can choose between detecting more fraudsters, including a higher rate of false positives or catching less fraudsters, but increase Precision and therefore avoid false positives. This choice can depend on the ability of the merchant to deal with false positives and on the merchants specific total fraud costs. In the context of fraud detection the term total fraud costs means the sum of lost value, scanning cost as well as reimbursement fees associated with a fraud case.

After a certain threshold value, in the shown case 12, the Precision is almost 1 and will only increase insignificantly. The Recall and consecutively F1, will decrease from that point. The Reason for this is the intrinsic mechanic in the DNA approach. Fraudulent transactions with a comparable low fraud profile will be assigned a lower risk level. This however, is still higher level than the risk level of genuine users. If however, the threshold is set high enough these lower profile fraud cases will be classified incorrectly as genuine, causing the Recall and F1 to drop. Therefore it makes no sense to choose a threshold greater than 12.

IV. EXPERIMENTAL EVALUATION

The performance of the proposed DNA approach is compared to the standard techniques, mentioned in the related work. This comparison is based on real credit card fraud data, which was thankworthy provided by a successful gaming company on the online games market.

A. Data Set

The given data set (referred as full data set) comprises of 156,883 credit card transactions from 63,933 unique users. The records in the data set have the schema as it can be seen in Table II. Due to the high number of occurrences in several columns as well as the lack of distinctive attributes, most of the standard techniques were not applicable on that data set. To overcome these obstacles and to get a fair comparison, several adaptations to the data set have been done. The resulting prepared data has a minimum sequence length of three (smaller sequences have been discarded) and four derived attributes (see Table III) were added.

TABLE II. FULL DATA SET SCHEMA

column Name	description
created	timestamp of the payment transaction
user_signuptime	
creditcard_token	identifies credit card, hashed
card_bin	Bank Identification Number
user_country	user´s land of origin
user_id	
user_email	hashed for privacy compliance
transaction_amount	
order_payment_status	

TABLE III. ADDITIONS PREPARED DATA SET

column Name	description
bin_country	2 letter country code derived from card_bin
days_since_signup	integer attribute calculated as difference from signup_time to created
total_count	denotes total transaction figure for a particular user_email
package	a single letter attribute ranging from A to E. It was derived from the offer_price attribute to reduce the cardinality of the offer_price attribute

The prepared data set comprised of 13,298 unique users which are accompanied by 46,516 transactions. The last transaction of each user was cut out in order to form the test data set. This procedure segmented the prepared data set into 71.4 % train data and 28.58 % test data.

B. Fraud Detection Performance

All tests in this section were performed using the prepared data set. We used the F1 score in order to rank the compared methods. As shown in Figure 4, the DNA approach is able to perform 16.25 % better than the best standard method, which is the Bayesian Net.

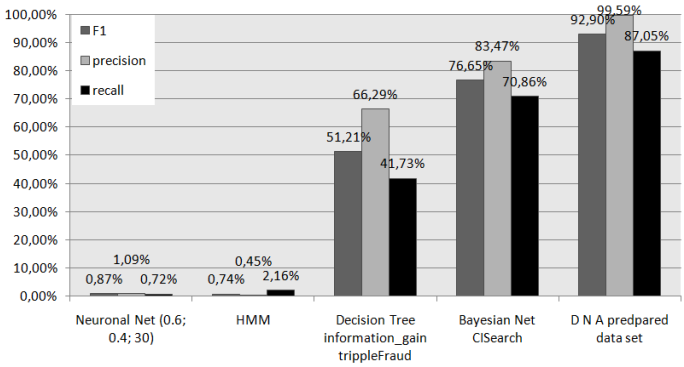


Figure 4. Fraud detection performance comparison

The DNA approach is also able to achieve an almost perfect 99.59 % Precision, which is especially valuable for online gaming merchants, since it reduces the risk of punish

genuine users and consecutively reduces the risk of reputation loss.

C. Scalability

The previous subsection showed the impressive fraud detection accuracy of the DNA approach. The next step was to take a deeper look into the run time behavior of the compared methods. Therefore, we prepared scaled data sets by multiplying the original prepared data (Subsection IV A) by 4, 8 and 16 times. The corresponding total execution times in seconds of the used methods can be seen in Figure 5.

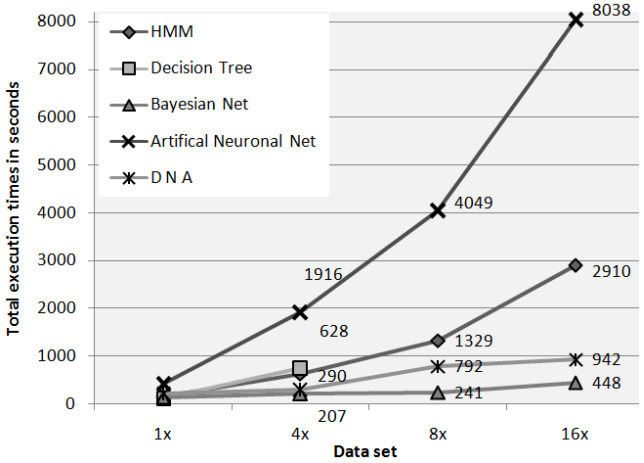


Figure 5. Execution time of all implemented methods (the used DTs produced out of memory exceptions for data set 8x and 16x)

The DNA approach uses simple operations and data structures like HashMaps or simple additions for calculating the risk level per sequence. This lowers the computational intensiveness and improves scalability significantly. The Bayesian Belief Networks obtain consistently the best results for static analysis (all transactions are classified in one batch). However, the BBN is an offline algorithm; therefore it cannot be applied in a real time online payment system on a transaction basis. To use offline algorithms in real applications, further timeslots need to be considered which were neglected in this work (data collection, transfer and preparation time).

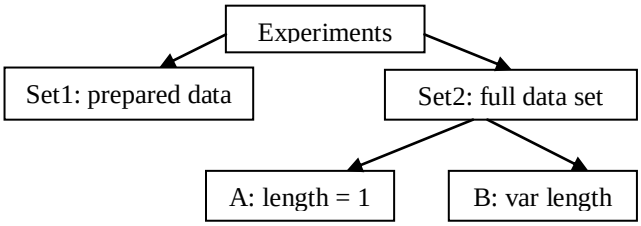


Figure 6. Overview experimental evaluation

The second set of experiments measures the execution time of the DNA approach using the full data set and varying the sequence length. Multiplications of the full data set (Section IV, A) were used for these experiments. In order to perform the measurements, the algorithm was adapted to record the execution time per sequence. This enables us to measure the average execution time for each sequence length.

Constant Sequence Length: In the first round of experiments, we measured the execution time for the fixed sequence length = 1. This allows measuring the influence of the data set size on the execution time.

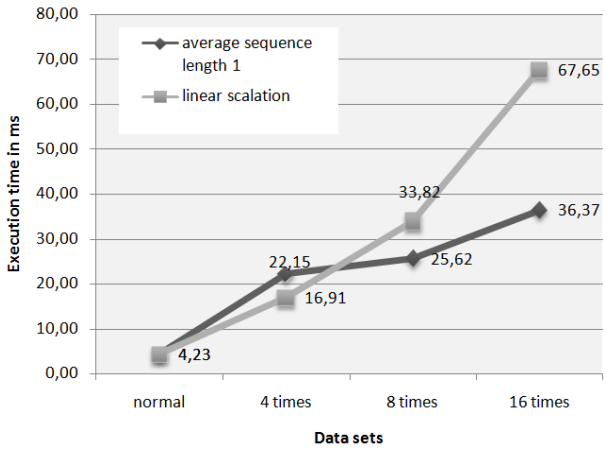


Figure 7. DNA results for fixed length sequence

In Figure 7, the grey line indicates the linear scaled value, based on the results of the normal data set. This means that a linear scaling algorithm, which needs 4.23 [ms] for a sequence of length 1 on the normal table, would need 16 times as long for the same calculation on a 16 times bigger table. As it can be seen the execution time for the DNA approach is less than that.

Higher Sequence Length: Throughout the second round of experiments the average execution times for all sequence lengths were averaged. Since the above mentioned data sets were created by multiplication of the full data set, not only the data set size but also the sequence lengths were increased. Figure 8 shows the experimental results.

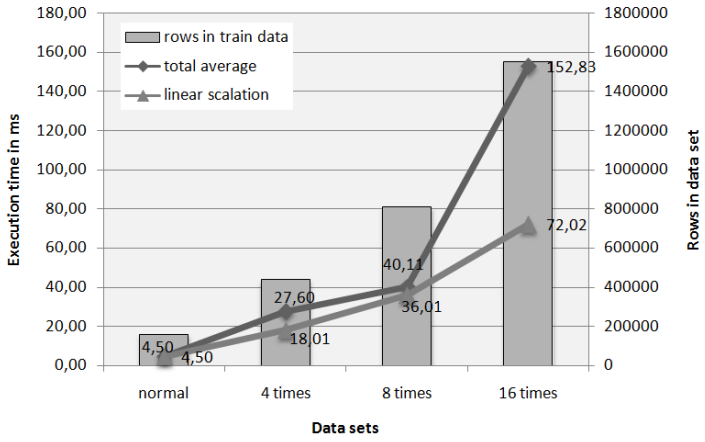


Figure 8. DNA results over higher sequence length

It is visible that the performance stays close to the expected linear response time behavior for the first three data

set sizes. There is a knee-point somewhere between the 8th and the 16th size of the data set, on which the execution time increases over proportionally. These results indicate that the sequence length has a greater influence on the execution time, than the data set size. This property does not necessarily impose a problem for online games merchants since the sequence length usually stays well below critical length.

All performance numbers presented in the experiment evaluation (Figure 7 - 8) reflect the pure response time of the DNA. The execution times of the other algorithms mentioned in Figure 5, display only the decision time of the trained algorithms. An additional overhead has to be taken into account, which reflects time for data collection, data transfer and preparation time. Hence, in practical terms, the reported performance delta is conservative. In real scenarios it will be higher.

V. CONCLUSION

This work deals with the problem of fraud detection in online games. The problem is caused by the lack of useful financial data, the anonymity in online games as well as the comparably short transaction sequences. The problem is solved by introducing an algorithm that is able to find and use distinctive attributes within sequences. In addition, a concept of country clusters is used to evaluate the legitimacy of a transaction. The DNA approach performs 16.25 % better than the best standard method (Bayesian Net) and achieves 99.59 % Precision. The achieved Recall rate (87.05 %) reduced the probability for false negatives and therefore the need for human intervention is reduced. In addition, the DNA approach scales better than other approaches with increasing data volumes, while offering acceptable response/detection times. This allows the application of the DNA approach in a real time online fraud detection system.

Future Work: The algorithm suggested in this work is especially designed to overcome the limiting conditions given in the online fraud detection area. The experiments showed good fraud detection results. However, further development is needed in order to reduce the influence of the sequence length on the execution time. It would also be helpful to incorporate the sequence length into the algorithm. The algorithm may be susceptible to the sequence length due to the proposed additive technique depicted in (4). The used data set did not allow us to precisely quantify possible impacts. Another direction of development could be the abstraction of the proposed attribute construction technique and their application on other classification domains.

REFERENCES

[1] n.n. Video Games Wiki http://vgsales.wikia.com/wiki/Video_game_industry [retrieved: June, 2013]

[2] n.n. U.S. Leads the World in Credit Card Fraud Report http://www.businesswire.com/news/home/20111121005121/en/U.S.-Leads-World-Credit-Card-Fraud-states [retrieved: June, 2013]

[3] J. Han, M. Kamber and J. Pei, "Data mining: Concepts and techniques" 3. edition, 2012

[4] S. Ghosh, and D. Reilly, "Credit card fraud detection with a neural-network" In: Proceedings of the Twenty-Seventh Hawaii International Conference, 1994, p. 621–630

[5] J. R. Dorronsoro, F.Ginel, C. Sgnchez and C. S. Cruz, "Neural fraud detection in credit card operations" In: IEEE Transactions on Neural Networks 8 (1997), No. 4, p. 827–834

[6] R. Brause, T.Langsdorf and M. Hepp, "Neural Data Mining for Credit Card Fraud Detection", 2000

[7] S. Maes, K. Tuyls, B. Vanschoenwinkel and B. MANderjack, "Credit Card Fraud Detection Using Bayesian and Neural Networks"In: Proceedings of NF2002, 2002

[8] K. Ezawa and S. Norton, "Constructing Bayesian networks to predict uncollectible telecommunications accounts" In: IEEE Experts (1996), Vol 11, Issue 5, p. 45–51

[9] W. Lam and F. Bacchus, "Learning Bayesian Belief Networks: An approach based on the MDL Principle" In: IJCI Vol 13, 1994, pp. 269–293

[10] M. Gadi, X. Wang and A. P. Lago, "Comparison with Parametric Optimization in Credit Card Fraud Detection" In: Seventh International Conference on Machine Learning and Applications, 2008, p. 279–285

[11] A. Srivastava, A. Kundu, S. Sural and A. K. Majumdar, "Credit Card Fraud Detection Using Hidden Markov Model" In: IEEE Transactions on Dependable and Secure Computing 5, 2008, No. 1, S. 37–48.

[12] S. Mhamane and L. M. R. J. Lobo, "Fraud Detection in Online Banking Using HMM" In: International Conference on Information and Network Technology (ICINT 2012), 2012, Vol. 37.

[13] V. Bhusari and S. Patil, "Application of Hidden Markov Model in Credit Card Fraud Detection" In: International Journal of Distributed and Parallel systems 2, 2011, No. 6, p. 203–211

[14] S. Dhok, "Credit Card Fraud Detection Using Hidden Markov Model" In: International Journal of Soft Computing and Engineering (IJSCE), 2012, Vol. 2

[15] L. Rabiner and B. Juang, "An introduction to hidden Markov models" In: IEEE ASSP Magazine 3,1986, No. 1, p. 4–16.

[16] M. Stamp, "A Revealing Introduction to Hidden Markov Models", 2012, URL http://www.cs.sjsu.edu/~stamp/RUA/HMM.pdf [retrieved: June, 2013]

[17] J. R. Quinlan, "C4.5: Programs for machine learning", 1998

[18] T. Minegishi and A. Niimi, "Detection of Fraud Use of Credit Card by Extended VFDT", 2011

[19] Y. Sahin and E. Duman, "Detecting Credit Card Fraud by Decision Trees and Support Vector Machines" In: Proceedings of the International MultiConference of Engineers and Computer Scientists, 2011, Vol. I

[20] K. K. Sherly and R. Nedunchezhian,"BOAT adaptive credit card fraud detection system", 2010

[21] M. F. A. Gadi, X. Wang and A. P. Lago, "Comparison with Parametric Optimization in Credit Card Fraud Detection" In: Seventh International Conference on Machine Learning and Applications, 2008, p. 279–285.

 ISBN: 978-1-61208-295-0

The Data Checking Engine: Monitoring Data Quality

Felix Heine, Carsten Kleiner, Arne Koschel
University of Applied Sciences & Arts Hannover
Faculty IV, Deptartment of Computer Science, Hannover, Germany
Email: firstname.lastname@hs-hannover.de

Jörg Westermayer
SHS Viveon
Germany
Email: joerg.westermayer@shs-viveon.de

***Abstract*—In the context of data warehousing and business intelligence, data quality is of utmost importance. However, many mid-size data warehouse (DWH) projects do not implement a proper data quality process due to huge up-front investments. However, assessing and monitoring data quality is necessary to establish confidence in the DWH data. In this paper, we describe a data quality monitoring system developed collaboratively at HS Hannover and SHS Viveon: The "Data Checking Engine" (DCE). The goal of the system is to provide DWH projects with an easy and quickly deployable solution to assess data quality while still providing highest flexibility in the definition of the assessment rules. It allows to express complex quality rules and implements a template mechanism to facilitate the deployment of large numbers of similar rules.**

Keywords*—*Data Quality, Quality Rules, Data Analysis, Data Quality Monitoring, Data Warehouses

I. INTRODUCTION

Data quality (DQ) is of utmost importance for a successful data warehouse project. In this context, continuous monitoring is an integral part of any DQ initiative. In this paper, we describe a data quality monitoring system called *Data Checking Engine* (DCE) developed collaboratively at the University of Applied Sciences & Arts Hannover and SHS Viveon. The main goal is to provide a flexible, yet simple tool to monitor data quality in DWH projects, which can also be used during the DWH development to test its Extract Transform Load (ETL) process.

To constantly monitor the quality of data of a database, it is necessary to define quality rules using a flexible rule definition language. Quality rules are either derived from business rules or found via profiling or data mining. They are executed either in regular intervals or based on specific events like the completion of an ETL job. The results of the rule runs are recorded in a result repository, which also keeps historical data so that users can evaluate the quality of data over time. As rules will evolve over time, it is necessary to keep a history of rule definitions so that historic results can be related to the correct version of the rule's definition.

We believe the ability to express complex rules is crucial. A set of hard-coded rule types found in some data quality tools is typically only suitable to detect rather simple quality problems on the attribute or single tuple level. However, there are more complex data quality problems, which cannot be detected using such rules. As an example, consider an error located in the logic of an ETL process. Due to this error, the process fails to reference the correct product group for some of the records of a sales fact cube. The bug is subtle and does not show up very often. At the attribute level all sales records are correct. However, the trend of the time series showing the sales sum with respect to individual product groups will indicate a quality problem.

It requires skilled users to write such rules but larger sets of rules will look similar in structure. They differ only in the tables and attributes they are applied to. For this, a template mechanism is useful to help users define such rules. The idea is that only the template writer has to cope with the full complexity; template users can then apply these templates to their tables and attributes.

To avoid discontinuity of the reporting environment for DWH users, re-using existing Business Intelligence (BI) tools is superior over building a specialized quality reporting GUI. Still it is sufficient to export rule results to a quality data mart, which can then be accessed by any standard BI tool. However, the plain rule results have to be aggregated to more comprehensive quality metrics in a flexible and user defined way.

Furthermore, the rules themselves have to be tested in the development environment before deployment. Thus, an automated transfer and synchronization with the production system is necessary.

In a nutshell, we target the following requirements:

- Express complex rules
- Reduce complexity of rules (utilizing a template mechanism)
- Execute the rules regularly or upon specific events
- Keep a history of rule definitions and execution results
- Store this history in a quality data mart persistently
- Aggregate the rule results to quality metrics
- Provide export/import mechanism for rule meta data

The remainder of this paper is organized as follows: In the following section, we give an overview of related work. Section III focuses on the definition of quality rules and explains our template mechanism. Section IV describes the DCE architecture. In the subsequent section, we briefly elaborate on quality metrics. In the final section, we summarize our achievements and give an outlook to our future plans.

II. RELATED WORK

Over the last decade, much research in the data quality domain has been conducted, see for example [1]. Research areas related to data quality are outlier detection, data deduplication, data quality monitoring, data cleansing, and data mining to detect quality rules. We are specifically interested in monitoring and reporting data quality. In general, we follow

 ISBN: 978-1-61208-295-0

the approach of Kimball [2] who outlines an approach to DQ assessment in DWH systems.

For our work, we are especially interested in formalisms to describe quality rules. Most existing approaches target only specific types of rules. Edit rules describe invalid tuples [3]. In [4], editing rules are used to match records with master data. In [5], *conditional functional dependencies* are used to express more complex rules spanning multiple tuples of a relation. The same book also describes *conditional inclusion dependencies* that are generalizations of referential integrity checks. These approaches can be reformulated to SQL, thus DCE is able to execute such rules. Another type of rules are differential dependencies, see [6].

In the domain of data deduplication (also called record linkage), rules are important to describe matching criteria. As an example, the IntelliClean [7] system uses rules like `<if> condition <then> action with probability p` to match duplicates. Currently, we do not target this issue, although we plan to integrate these features in our system in the future.

Another approach is to extend SQL to incorporate data quality features. An example is the FraQL [8] language that specifies pivoting features and allows to integrate user defined grouping and aggregate functions that allow to analyze data more comfortably. The drawback is that a special execution engine is required. Thus, the features of existing relational optimizers are not available or have to be reproduced.

Furthermore, many prototypic research systems and commercial tools are present. For an overview, see [9]. Most existing tools focus on dimension data only and thus stress single record problems and deduplication.

However, to the best of our knowledge, no tool provides a similar mechanism that allows to build complex rule templates, which can, for example, be used to test indicator values against time series models.

III. Rule Definition Language

A central issue is the language to define the quality rules. On one hand, it has to be expressive to allow for complex rules like time series tests. On the other hand, fast definitions of simple rules like NULL value checks has to be possible. Also, the rule execution is typically critical with respect to execution time and resource consumption. As large datasets have to be checked, an efficient rule execution engine is demanded.

Thus, we decided to rely on the native SQL executor of the DBMS. This means, the core of each rule is an SQL statement, which collects the required information from the underlying tables. This statement is written by the DCE user, allowing even vendor-specific optimizations like optimizer hints.

DCE defines a *standard attribute set* for the result tuples. The rule statements have to adhere to this standard. Each statement computes a *result value*, which is the basis for the rule check. For a NULL rule, the result value might be the percentage of NULL values of the checked values. There might either be a single result value or multiple values, broken down by dimensional hierarchies. The latter case might for example yield a percentage of NULL values for each product group in each region. For each rule, *multiple bounds* can be defined, specifying valid ranges for the observed values. The bounds can be activated or deactivated with respect to all values contained in the result tuple. In this way, the bound for NULL values can be normally defined to be 5 percent, however for specific product groups it might be higher. A specific application for this feature is to change bounds for business metrics, e.g., according to the week day. Typically, the revenue sum for traditional stores might be zero on Sundays.

A *severity* can be assigned to each rule bound, and multiple bounds with different severity can be defined for a rule. The severity information of failed rules is returned to the scheduler. Based on this information, the scheduler might, e.g., decide to interrupt an ETL process or to alert the DWH team.

Each rule's SQL statement can have *multiple parameters*, which are set at execution time. These parameters can for example be used to determine the range of data to be checked. In this way, a quality rule running after an ETL job might be limited to check only the new records in a fact table.

In the following, we describe a *sample rule*. The basic idea of the example is to test indicator values like revenue stored in a fact table on a regular basis against a moving average of its past values. For simplicity, we assume there is no seasonal component, although this is not a limit of the system. The following formula describes our model:

$$Y_t = \frac{1}{k}\sum_{d=1}^{k} Y_{t-d} + \epsilon_t \tag{1}$$

Here, ϵ_t is a random component, which is Gaussian with $\mu = 0$ and some σ. The trend is based on the moving average of the k previous values of the indicator.

During DWH operation, we assume that past values for the indicator are already checked and corrected. Each day after the ETL process has finished, we want to test the new value. Thus, we have to calculate

$$k_t = Y_t - \left(\frac{1}{k}\sum_{d=1}^{k} Y_{t-d}\right) \tag{2}$$

and then check whether k_t is within a certain interval. As ϵ_t is Gaussian, we know that 95% of the values will be within the interval $[-2\sigma, 2\sigma]$. We could use these bounds to generate a warning and the $[-3\sigma, 3\sigma]$ interval to generate an error.

```
SELECT today.day day,
       avg(today.revenue) -
       avg(past.revenue) result
FROM sales_fact today, sales_fact past
WHERE today.day = $testday$
  AND (today.day - past.day)
      BETWEEN 1 AND k
GROUP BY today.day
```

Fig. 1. Sample rule code (simplified)

A simplified SQL for these checks is shown in Fig. 1. The statement returns a result value k_t, which is then checked by the DCE against bounds like the $[-2\sigma, 2\sigma]$ interval. This statement has a parameter `$testday$`, which is replaced at runtime with the current day.

 ISBN: 978-1-61208-295-0

In typical environments, there is often a need to define a number of equivalent rules over a large number of tables and attributes. To accommodate for this requirement, we implemented a *template concept*.

A template looks quite similar to a normal rule. It contains a SQL statement producing the same set of standard columns, and it might also contain bound definitions. However, instead of the target table and attribute names, the template's SQL statement contains special markers. For attributes, these markers declare the purpose of the attribute within the rule. Once the user has defined a template, she can instantiate it for multiple sets of tables and attributes. During this process, she either defines new bounds or uses the predefined bounds from the template for the generated rules. The engine forwards rule parameters defined within the template to the generated rules.

```
SELECT today.§refdim1§ day,
       avg(today.§refattr1§) -
       avg(past.§refattr1§) result
FROM §reftable1§ today, §reftable1§ past
WHERE today.§refdim1§ = $testday$
  AND (today.§refdim1§ - past.§refdim1§)
      BETWEEN 1 AND k
GROUP BY today.§refdim1§
```

Fig. 2. Sample template code (simplified)

The *sample* statement is a good candidate for a template. In the template, there is another type of parameters called template parameters that are replaced at template instantiation (i.e., rule creation time). These are used to define placeholders for the table and attribute names, like §reftable1§ (cf. Fig. 2).

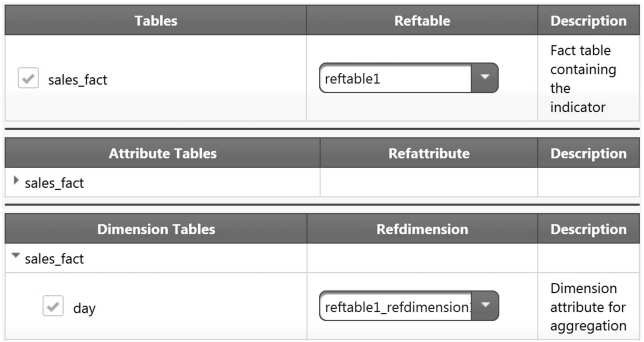


Fig. 3. Instantiating a template

A *GUI* assists unexperienced users with defining the template parameters, as shown in Fig. 3. For this dialog, the GUI reads the database catalog and lets the user map the template parameters to catalog objects. E.g., §reftable1§ is replaced with sales_fact.

IV. ARCHITECTURE

Figure 4 shows an overview of the DCE overall architecture. The DCE itself is organized as a classical three-tier application. It interacts with the enterprise data warehouse system in order to compute quality indicators. Also, results of the data quality checks may be propagated into another external database system, the data quality data mart. This database in itself is also organized as a data mart and provides long term storage of computed data quality indicators in order to be used for long term analysis of enterprise wide data quality. In a sense it is a meta-data warehouse for data quality. There is also an external scheduling component (typically standard system scheduling capabilities), which triggers computation of data quality indicators at previously defined points in time.

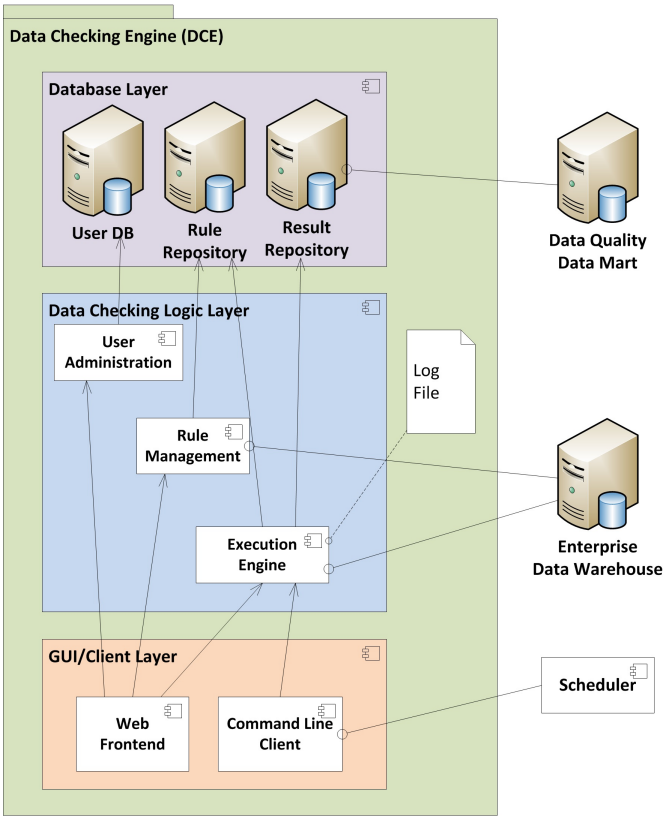


Fig. 4. Data checking engine architecture overview

Within the DCE itself the main entry point for data quality managers is the GUI of the DCE web application (shown at the bottom of Fig. 4). The GUI is used to manage users of the DCE application, to manage data quality rules, and to manage data rule executions. As typically the execution of data quality checks is not triggered manually, there is also a command-line client library for the rule execution engine that is triggered by an external scheduler. The schedule to be used is managed in the web application as well.

The main data checking business logic can be found in the middle tier. This logic is used by the web application as described above. Note that there is a strict separation between user management, rule management and rule execution management in the middle tier as well. Whereas the user administration component provides standard functionality, note that the rule management component contains advanced features. For instance the template mechanism described in the previous section is implemented here.

The execution engine is also managed by the web application: on one hand rules can be manually executed from the web application, on the other hand scheduled execution can be defined here.

 ISBN: 978-1-61208-295-0

During rule execution, the engine replaces the parameters in the rule's SQL statement with their current values and then runs the statement using the target database. Thus, moving large amounts of data into the DCE engine is avoided. The result of the SQL statement is then further processed. This includes checking the currently applicable bounds and testing their severity.

In the execution engine it is also defined, which rules are executed on what data warehouse database under whose privileges. Note that multiple different data warehouses (or database systems) may be used as source, because the connection information is also managed by the web application.

Finally the database layer consists of three separate areas:

- Rule repository, which holds the data quality rules as well as base templates
- Result repository holding results of rule execution
- User database which is used for access management to only the DCE itself

Once results of the executed data quality rules have been stored in the result repository they may be propagated to the data quality data mart that aggregates the results into quality indicators.

This data mart is not part of the DCE but located within the standard DWH infrastructure of the company. Thus, standard interfaces such as reporting and BI tools can be used to further present and analyze the data quality status. This way the additional effort for data quality monitoring can be kept minimal as access to data quality indicators follows well established processes and uses well-known tools, which are used for regular monitoring of enterprise performance indicators as well. In addition, the concept of viewing data quality indicators similarly as regular performance indicators is very fitting, as these have to be tracked accordingly in order to ensure reliability of data in the data warehouse. Ultimately, this is necessary to make the right entrepreneurial decisions based on reliable information.

V. Data Quality Metrics

The result repository contains large amounts of specific results that individually describe only a very small fraction of the overall data quality of the DWH. In order to get a quick overview of the quality level, a small set of metrics that aggregate the rule results is required.

In the literature, there are various approaches to define data quality indicators, for example [10]. Thus we decided to provide a flexible approach that enables the user to define her own indicator hierarchies. The engine stores indicator definition meta data and calculates the resulting indicator values.

An important issue here is to take incremental checks into account. As an example, consider a rule that checks the number of dimension foreign keys in a fact table that reference a dummy instead of a real dimension entry. As the fact table is large, the daily rule just checks the new fact records loaded in the previous ETL run. Thus the indicator has to aggregate over the current and past runs to provide an overall view of the completeness of the dimension values.

VI. Conclusion and Future Work

We have already implemented a prototypical version of the described architecture. Furthermore, we have validated the approach by interviewing teams of different DWH projects and building project specific prototypical setups. Our engine has been able to support their quality monitoring requirements. Especially the flexibility in rule definition was appreciated. We have not only detected quality problems on tuple level but also more complex issues, e.g., checking the trend of indicators stored in a fact table. As expected, our template mechanism has proven to be an important way to simplify rule definition.

The engine keeps a comprehensive history of rule results *and* rule meta data, which allows to monitor data quality over time and to check whether quality improvement projects were successful. This quality data is exposed to external BI tools for reporting and further analysis.

An important consequence of the flexibility of our approach is that the DCE can also be used during DWH/ETL development to test the result processes. The testing rules developed during this project phase may also be used during normal operation, reducing the overall cost.

Our approach is currently working on any relational database system. In the future, we plan to also integrate Big Data systems like Hadoop, as more and more relevant data is stored in such systems. Thus data quality should be monitored there as well. As there is currently no universal query language standard like SQL in the relational sector, we will have to devise a flexible way to cope with various rule definition languages.

References

[1] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[2] R. Kimball and J. Caserta, *The data warehouse ETL toolkit*. Wiley, 2004.

[3] I. P. Fellegi and D. Holt, "A systematic approach to automatic edit and imputation," *Journal of the American Statistical Association*, vol. 71, no. 353, pp. 17–35, 1976.

[4] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu, "Towards certain fixes with editing rules and master data," *The VLDB Journal*, vol. 21, no. 2, pp. 213–238, Apr. 2012. [Online]. Available: http://dx.doi.org/10.1007/s00778-011-0253-7

[5] W. Fan and F. Geerts, *Foundations of Data Quality Management*, ser. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2012.

[6] S. Song and L. Chen, "Differential dependencies: Reasoning and discovery," *ACM Trans. Database Syst.*, vol. 36, no. 3, pp. 16:1–16:41, Aug. 2011. [Online]. Available: http://doi.acm.org/10.1145/2000824.2000826

[7] M. L. Lee, T. W. Ling, and W. L. Low, "Intelliclean: A knowledge-based intelligent data cleaner," in *ACM SIGKDD, Boston, 2000*, 2000.

[8] K. Sattler, S. Conrad, and G. Saake, "Adding conflict resolution features to a query language for database federations," in *Proc. 3nd Int. Workshop on Engineering Federated Information Systems, EFIS'00, Dublin, Ireland, June*, 2000, pp. 41–52.

[9] J. Barateiro and H. Galhardas, "A survey of data quality tools," *Datenbank-Spektrum*, vol. 14, 2005.

[10] B. Heinrich, M. Kaiser, and M. Klier, "Metrics for measuring data quality - foundations for an economic oriented management of data quality," in *Proceedings of the 2nd International Conference on Software and Data Technologies (ICSOFT). INSTICC/Polytechnic Institute of Setúbal*, J. Filipe, B. Shishkov, and M. Helfert, Eds., 2007.

 ISBN: 978-1-61208-295-0

Exploiting Wiktionary for Lightweight Part-of-Speech Tagging for Machine Learning Tasks

Mario Zechner, Stefan Klampfl, and Roman Kern
Know-Center GmbH
Graz, Austria
Email: {mzechner,sklampfl,rkern}@know-center.at

***Abstract*—Part-of-speech (PoS) tagging is a crucial part in many natural language machine learning tasks. Current state-of-the-art PoS taggers exhibit excellent qualitative performance, but also contribute heavily to the total runtime of text preprocessing and feature generation, which makes feature engineering a time-consuming task. We propose a lightweight dictionary and heuristics based PoS tagger that exploits Wiktionary as its information source. We demonstrate that its application to natural language machine learning tasks considerably decreases the feature generation runtime, while not degrading the overall performance on these tasks. We compare the lightweight tagger to a state-of-the-art maximum entropy based PoS tagger in clustering and classification tasks and evaluate its performance on the Brown Corpus. Finally, we explore future research scenarios where our tagger and Wiktionary lookup enables efficient processing of big data due to the significant decrease in runtime.**

***Keywords*—Machine learning; feature engineering; natural language processing; part-of-speech tagging; big data.**

I. INTRODUCTION

Large scale corpora of unstructured natural language text, as found on the web or in enterprise document management systems, are common application fields for various supervised and unsupervised machine learning algorithms. The goal in most of these scenarios is to extract structured information from the unstructured text. This can include global structure derived from clustering, local structure as detected by topic segmentation, or meta-data such as authors, named entities, facts, or genre.

In many of these scenarios, the unstructured text is first preprocessed and transformed into a suitable feature representation for the machine learning algorithm used. A common feature representation is the classical text vector space or bag-of-words model [1]. In this space, each term in the corpus is represented by an own dimension. A text is transformed into this vector space by counting the frequency of each term it contains and storing these frequencies in the corresponding components of the vector. In order to counteract certain artifacts, such as differences in the length of documents, these raw term-frequency vectors are often additionally weighted using various schemes such as term frequencyŰinverse document frequency (TF-IDF) [2] or BM25 [3].

In addition to weighting term vectors, terms can be pruned by not taking them into account when transforming texts to term vectors. Specific word categories do not transport the concepts or meanings of a text and are thus often omitted from the feature representation. One way to prune these words is via stop-word lists [4], which are usually non-exhaustive. A more sophisticated approach is part-of-speech tagging, which assigns a word category such as noun, verb or adjective, to each term in the document. Instead of specific words, entire word categories are omitted, which are unlikely to encode concepts, such as determiners, particles, verbs and so on. Nouns and proper nouns are the most likely word categories to transport meaning, and it is usually terms from these categories that get included into the feature representation of a text document.

Part-of-speech (PoS) tagging has been an active research topic over the last two decades. Various approaches have been devised, from rule-based systems [5], to different statistical approaches via Hidden Markov Models [6], Maximum Entropy Models [7], Conditional Random Field (CRF) models [8] or Support Vector Machines (SVM) [9]. Especially the statistical approaches exhibit excellent accuracy. However, these approaches also incur a considerable increase in runtime, and often the feature extraction based on PoS tags can take longer than the actual task at hand. Moreover, feature engineering becomes more cumbersome as modifications of and experimentation with the feature engineering strategy necessitates a rerun of the feature extraction stage.

For big data scenarios arising from web corpora, this increase in runtime is a big hindrance. Processing times can only be decreased at best in a linear manner by adding more machines to solve the problem. In research scenarios hardware budgets are bounded, so a different strategy that decreases the time complexity of tagging is preferable.

Large scale corpora as described above are not only problematic due to their size, but also due to the fact that they are often multi-lingual. While English PoS tagging models are available, PoS tagging models for other languages are harder to obtain. This can be attributed to the lack of training corpora through which statistical PoS tagging models can be trained. Therefore, a tagging system that does not rely on training corpora would be desirable.

We hypothesize that a dictionary and heuristics based tagging approach is sufficient in quality for the above described application scenarios if its recall is comparable to that of more sophisticated methods. Furthermore, this tagging approach should have a considerable edge over more sophisticated approaches in terms of runtime performance. In this paper, we exploit Wiktionary [10] as a freely available, multi-lingual information source which allows us to tackle the problem in an efficient and cheap manner.

 ISBN: 978-1-61208-295-0

Contribution

Our contribution consists of the following:

1) A lightweight, dictionary and heuristics based PoS tagger based on Wiktionary, that is fast, sufficiently accurate and cheaply adaptable to other languages.
2) An evaluation of PoS tagging and runtime performance on the Brown Corpus relative to a state-of-the-art tagger, allowing us to estimate its performance when used for the feature engineering stage of machine learning tasks.
3) An evaluation of the PoS tagger as part of the feature engineering stage of a text clustering task, showing that using the lightweight tagger decreases the overall runtime of the scenario considerably, while retaining the same quality as achieved with the state-of-the-art tagger.
4) An evaluation of the PoS tagger as part of a text classification task, again showing that the lightweight tagger decreases the overall runtime while retaining the same quality as achieved with the state-of-the-art tagger.
5) A discussion of potential applications and implications on big data tasks.

II. WIKTIONARY-BASED PART-OF-SPEECH TAGGING

Our lightweight PoS tagger uses Wiktionary as an information source. Wiktionary is a freely available, multilingual dictionary, thesaurus, and phrase book. It is edited by volunteers all over the world and currently contains dictionaries for 170 languages [11]. Wiktionary has an exhaustive list of criteria for inclusion of a word [12] and aims to capture common vocabulary. Proper nouns arising from person and company names, places and other named entities are included with specific caveats.

An article describing a term in Wiktionary generally contains information about a term's part-of-speech, word sense, pronunciation and so on. Usage examples are also often provided, as well as synonyms, antonyms, hypernyms and hyponyms. In addition, different spelling variations as well as reflections of a term are present.

This information varies across articles for one language, and between the collections of the 170 languages found in Wiktionary. The number of terms found in a language collection is also varying heavily depending on the language, from a few millions to a few dozens. Table I lists the top 10 languages and their individual article count at the time of writing. This shows that Wiktionary contains many large corpora for various languages. In this work we focus on English; however, performing the following experiments on corpora of other languages is easily possible, since building a lightweight tagger for a different languages involves parsing the corresponding articles in Wiktionary. This task is far less labour-intense than manually tagging a sufficient amount of training data.

As a first step we have built a parser for Wiktionary articles that extracts each word's forms, possible part-of-speech tags, synonyms, hyponyms, hypernyms and translations. For most words in Wiktionary, all its possible inflexions, e.g., "see", "saw", "seen" etc., are usually present. It should be noted that Wiktionary also contains multi-token phrases. For this work, we omit all such phrases.

Table I. THE TOP TEN LANGUAGES BY ARTICLE COUNT IN WIKTIONARY. THE LARGE AMOUNT OF ARTICLES AVAILABLE FOR DIFFERENT LANGUAGES MAKES OUR TAGGING APPROACH READILY APPLICABLE TO NON-ENGLISH LANGUAGES, WHICH IS NOT THE CASE FOR TRADITIONAL POS TAGGING METHODS.

Language	# articles
English	3,188,521
French	2,289,494
Malagasy	2,232,273
Chinese	828,580
Lithuanian	610,707
Russian	458,634
Greek	406,259
Korean	349,626
Swedish	329,137
Turkish	311,471

Table II. UNIFIED TAG SET USED BY OUR LIGHTWEIGHT TAGGER.

Tag	Examples
ADJECTIVE	green, valuable
ADVERB	strongly, quickly
CONJUNCTION	and, but, so
DETERMINER	the, an
NOUN	house, car
PROPER_NOUN	Stefanie, Linux
NUMBER	3.14, hundred
PARTICLE	who, whom
PRONOUN	his, theirs
PREPOSITION	in, from, on
PUNCTUATION	.;:!
VERB	see, fell, had
OTHER	anything else
UNKNOWN	special tag

The parsed information is then fed into an index that is easily queriable. From this index, we derive the necessary information to build the lightweight tagger. We only use the forms and part-of-speech tags of a word. The tagger itself consists of a simple hash map with the lower cased word forms/inflexions as keys and the corresponding lists of potential PoS tags as values. The tags are normalized to a set of 14 word categories, given in Table II. We map other tagging systems such as the Penn Treebank tag set or the Stuttgart tag set to these 14 word categories as well. For brevity, we omit these mappings here, they can be found online [13] (user: anonymous, no password).

The annotation process consists of the following steps:

1) Tokenize the input text, e.g. via OpenNLP [14] or JTokenizer [15]
2) Transform each token to lower case, taking its locale into account
3) For each token, look it up in the hash map
   a) If found, return a random PoS tag from the list of the found entry
   b) If not found
      i) If the token does not start with a letter, return UNKNOWN
      ii) If the token starts with an upper-case letter, return PROPER_NOUN

 ISBN: 978-1-61208-295-0

iii) Else, return NOUN

This simple dictionary and heuristics based approach is equal to the baseline systems used in the evaluation of many statistical PoS tagger models. The handling of unknown words is motivated by the assumption that Wiktionary covers non-noun word groups exhaustively, while nouns and proper nouns are underrepresented. We therefore assume that any word not in the dictionary is a proper-noun, in case it starts with an upper case letter, and a noun otherwise. Tokens that do not start with a letter are tagged with UNKNOWN, indicating that the tagger has no information about what category this token belongs to.

III. EVALUATION

Our evaluation strives to provide empirical evidence for the following hypothesis:

- The lightweight tagger is comparable in recall to state-of-the-art taggers.
- The lightweight tagger is sufficiently precise.
- The lightweight tagger is considerably faster than the state-of-the-art tagger.
- The errors introduced do not negatively influence supervised and unsupervised machine learning computations.

The following sections describe experiments carried out to gather evidence for the above assumptions.

A. PoS Tagging

We evaluated the lightweight tagger's precision and recall on the English Brown Corpus [16], [17] and compared it to results obtained from the Maximum Entropy based PoS tagger in the freely available OpenNLP package, which was trained on the Penn Treebank corpus [18]. The Brown Corpus consists of roughly one million tagged words, from various genres such as news articles, editorials, humorous texts and so on. For the evaluation we mapped both the Brown Corpus tag set and the Penn Treebank tags emitted by the OpenNLP PoS tagger to our simplified tag set described in Section II. This allowed us to directly quantify the relative performance of our tagger relative to the OpenNLP PoS tagger.

We let both taggers tag the entire Brown Corpus and then calculated precision and recall based on the ground truth found in the corpus. We used the tokenization as given in the Brown Corpus instead of using a dedicated tokenizer. The results are shown in Figure 1, which shows the precision and recall of each tagger for specific word categories. Our focus is on nouns and proper nouns, as these are used to generate features for text classification and clustering. The lightweight tagger has similar recall for nouns compared to the OpenNLP tagger. The precision of the lightweight tagger for nouns is what one would expect for a baseline dictionary tagger. While not exactly stellar, it is still performing surprisingly well. For proper nouns, the OpenNLP tagger is clearly superior in both precision and recall. We attribute this to the fact that we only tag tokens that start with an upper case letter as proper nouns.

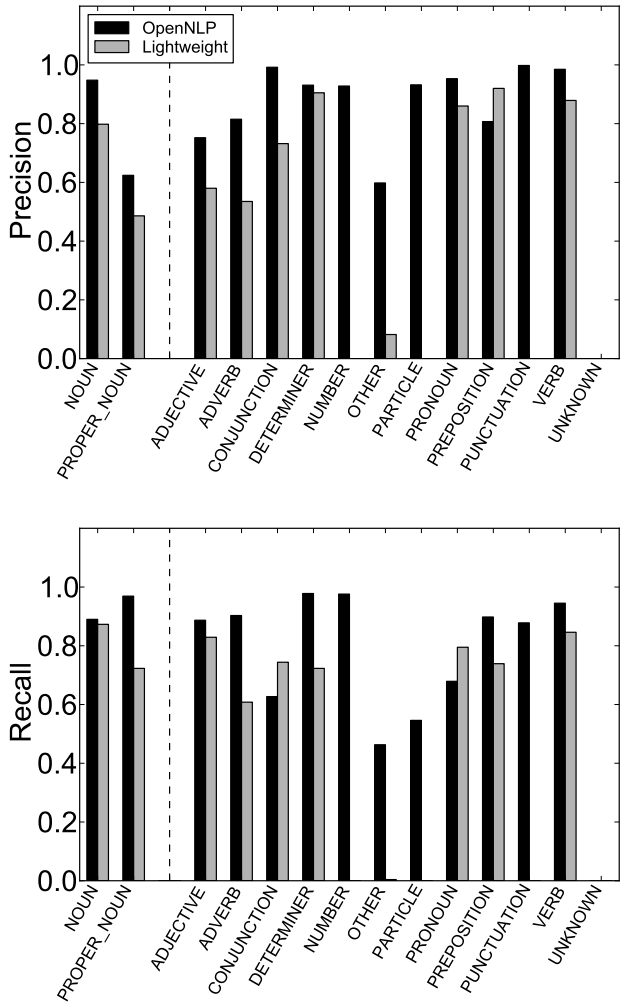


Figure 1. Comparison of precision (left) and recall (right) obtained by both the state-of-the-art tagger based on OpenNLP and our lightweight tagger on the tag set considered in this paper. For the relevant categories NOUN and PROPER_NOUN the lightweight tagger shows an acceptable precision and a recall comparable to the OpenNLP tagger.

We also measured the time spent on tagging. Our test machine was an Intel Core i7 CPU at 2.8Ghz, with 8GB RAM. Both the OpenNLP tagger and our lightweight tagger were written in Java. We ran the experiments in a 64-bit JVM from Oracle, version 1.7, update 9, assigning a maximum of 1GB of heap memory to the Java process. We ran each experiment 10 times and averaged the runtimes. The classification results are deterministic and where thus not averaged. Table III shows the average time taken by each annotator to annotate the entire Brown Corpus. As expected, the lightweight tagger was much faster than the much more sophisticated Maximum Entropy based tagger, which was outperformed by a factor of 100.

Another attribute of a dictionary and heuristics based tagger like the one presented here, is that it is more robust to grammatically incorrect text, found in social media or HTML pages cleaned and converted to plain text. The OpenNLP tagger, like other PoS taggers, needs sentence boundaries to function correctly. Our tagger has no such requirement and

 ISBN: 978-1-61208-295-0

Table III. TAGGING RUNTIME OF THE LIGHTWEIGHT TAGGER AND THE OPENNLP TAGGER FOR THE ENTIRE BROWN CORPUS AVERAGED ACROSS 10 RUNS. THE LIGHTWEIGHT TAGGER OUTPERFORMS THE MORE SOPHISTICATED MAXIMUM ENTROPY TAGGER BY A FACTOR OF 100.

Lightweight tagger	OpenNLP maximum entropy tagger
0.274s	34.219s

lends itself well to noisy text.

To establish whether the decrease in tagging performance has an impact on common natural text machine learning tasks, we evaluated both taggers in clustering and classification tasks as described in the next sections.

B. Feature Generation for Clustering & Classification

Our clustering and classification experiments share the same feature generation stage. Regardless of the corpus, we performed the following steps for each text document to transform it into a term vector. First, we tokenized each text document using the Maximum Entropy based tokenizer from the OpenNLP package. Each token was then stemmed using the Porter stemmer [19], [20] and normalized taking the English locale into consideration. We also applied a simple stop-word list found in Gate [21], filtering out high frequency words. Note that this step could actually be omitted since we are only taking nouns and proper nouns into account in the later stage.

For the OpenNLP tagger we also had to detect sentence boundaries with the corresponding Maximum Entropy sentence splitter in OpenNLP, which added additional processing time to the feature generation stage. Finally, we either tagged the tokens using the OpenNLP tagger or our lightweight tagger.

From the resulting list of tokens, we only took the stemmed and normalized form of nouns and proper nouns and converted them to term vectors as described in Section I. We then applied common TF-IDF weighting and normalized the vectors to unit length. In the TF-IDF scheme, the weight of term t in document d is given by

$$w_d(t) = \left(1 + \sqrt{n(t,d)}\right) \cdot \ln\left(1 + \frac{|D|}{n(t)}\right), \qquad (1)$$

where $n(t,d)$ is the number of times term t occurs in document d, $n(t)$ is the number of times the term t occurs in all documents, and $|D|$ is the total number of documents. These term vectors were then used as the input to our clustering and classification experiments.

C. Text Clustering

We evaluated our lightweight tagger in a text clustering scenario by comparing the achieved performance values to those obtained when using the OpenNLP tagger in the feature generation stage. We chose the 20 newsgroups corpus [22], [23] for our experiments as it is commonly used for text clustering evaluations. The 20 newsgroups corpus consists of roughly 20,000 documents, mostly equally collected from 20 different news groups, spanning topics such as atheism, politics, and sports. Each group is represented by around 1,000 documents. The corpus is split into a training and testing set for classification tasks, which we merged for this clustering scenario. We ran the experiment on both a manually selected subset of 5 groups ("rec.motorcycles", "alt.atheism", "talk.politics.guns", "comp.windows.x", "sci.crypt") as well as on the complete subset of all 20 newsgroups.

Table IV. PURITY AND CONDITIONAL CLUSTER ENTROPY AS WELL AS THE RUNTIMES OF FEATURE GENERATION (FE) AND CLUSTERING (C) ON THE SELECTED SUBSET OF 5 NEWSGROUPS. GOOD CLUSTERINGS HAVE HIGH PURITY AND LOW ENTROPY.

Configuration	Purity	Entropy	FE Runtime	C Runtime
All Tokens	0.3461	0.8928	13.678s	1.059s
NN/PN OpenNLP	0.8864	0.2897	29.090s	0.368s
NN/PN Lightweight	0.8915	0.2765	1.978s	0.378s

For clustering, we implemented the efficient online spherical k-means algorithm [24], using a constant learning rate of 0.01. We set the number of desired clusters to 5 and 20, respectively, and terminated the clustering algorithm after 3 iterations over the document set. For centroid initialization we used the k-means++ seeding strategy [25], which stochastically selects the initial centroids based on their distance to each other. In order to guarantee that all runs have the same initial conditions we provided a constant seed to the random number generator used. This enables a more effective comparison of different PoS taggers.

For each experiment we measured the purity and conditional cluster entropy of the resulting clusterings, as well as the runtimes of feature generation and clustering, averaged over 10 runs on our test machine described in Section III-A. Purity and conditional cluster entropy are both criteria for evaluating the clustering quality against a ground truth labeling [26]. Purity ranges between 0 and 1 and measures the accuracy of the assignment that is obtained if every instance would be labeled with the majority label within its corresponding cluster. A perfect purity of 1 is thus reached if all instances within each cluster have the same label. On the other hand, conditional cluster entropy measures the average amount of uncertainty about the cluster assignment of one instance that remains if its label is known. That is, a perfect clustering has a conditional cluster entropy of 0, since knowing the label completely determines the assigned cluster.

We performed the experiment on a total of 6 configurations, 3 runs for the selected subset of 5 newsgroups and another 3 runs on the full dataset of 20 newsgroups. The 3 configurations defined how the feature generation stage was carried out:

1) all tokens, stemmed, normalized,
2) nouns and proper nouns, stemmed, normalized, using the OpenNLP tagger,
3) nouns and proper nouns, stemmed, normalized, using the lightweight tagger.

Table IV summarizes the results on the 5 newsgroup subset, Table V describes the results on the full 20 newsgroup corpus.

The runtime of the feature generation stage was considerably lower for the lightweight tagger compared to the OpenNLP tagger. This was expected, though the relative speed-up is here 10 to 20-fold. This is a result of timing

 ISBN: 978-1-61208-295-0

Table V. PURITY AND CONDITIONAL CLUSTER ENTROPY AS WELL AS THE RUNTIMES OF FEATURE GENERATION (FE) AND CLUSTERING (C) ON THE FULL CORPUS OF ALL 20 NEWSGROUPS. GOOD CLUSTERINGS HAVE HIGH PURITY AND LOW ENTROPY.

Configuration	Purity	Entropy	FE Runtime	C Runtime
All Tokens	0.2659	0.7537	79.692s	12.833s
NN/PN OpenNLP	0.4432	0.5532	143.706s	4.145s
NN/PN Lightweight	0.4486	0.5541	7.300s	4.202s

Table VI. TOTAL NUMBER OF FEATURES GENERATED BY EACH CONFIGURATION, FOR BOTH THE SELECTED SUBSET 5 NEWSGROUPS AND THE FULL CORPUS

Configuration	# Features 5 NG	# Features 20 NG
All Tokens	99220	343408
NN/PN OpenNLP	28432	83388
NN/PN Lightweight	20842	63853

Table VII. COMPARISON OF THE NUMBER OF FEATURES OBTAINED WITH DIFFERENT FEATURE ENGINEERING METHODS ON BOTH THE TOTAL REUTERS CORPUS AND THE TRAINING SET ONLY. BOTH OUR LIGHTWEIGHT TAGGER AND THE STATE-OF-THE-ART OPENNLP TAGGER ONLY EXTRACT NOUNS AND PROPER NOUNS.

	training set	total
# documents	23,149	806,791
# features (all tokens)	49,427	303,732
# features (OpenNLP)	44,840	281,170
# features (lightweight)	42,049	255,147

Table VIII. COMPARISON OF THE RUNTIMES OF VECTORIZATION AND CLASSIFICATION OF THE REUTERS CORPUS IN DIFFERENT FEATURE ENGINEERING SCENARIOS.

	vectorization	classification
all tokens	382s	1810s
OpenNLP	1171s	1172s
lightweight	395s	1100s

the entire feature engineering stage instead of just the PoS tagging stage as described in Section III-A. Clustering times were comparable in case of the OpenNLP and lightweight tagger configurations, and higher in case of using all tokens. This can be explained by the fact that the clustering time is dominated by adjusting cluster centroids, which is proportional to the number of features in a cluster centroid. Using all tokens significantly increased the number of features of cluster centroids, as shown in Table VI.

As far as the quality of the resulting clusterings is concerned, using all tokens was highly detrimental to the clustering quality as shown in Tables IV and V. The additional tokens generated much noise, and the resulting cluster centroids were capturing features from all newsgroups. The other configurations for which only nouns and proper nouns were used performed considerably better. Both performed at approximately the same level on both the subset and the full corpus, supporting our hypothesis that using a sufficiently accurate PoS tagger like our lightweight tagger does not decrease clustering performance.

D. Text Classification

We evaluated our lightweight tagger also in a text classification scenario, by comparing the achieved performance values to those obtained with standard feature engineering methods. For this comparison we chose a classification task on the Reuters RCV1 corpus [27], a well-known dataset for document classification. It consists of 806,791 newswire stories that were collected over the period of one year, manually categorized, and made available by Reuters Ltd for research purposes. The dataset was labeled with respect to three different sets of categories: *Topics*, *Industries*, and *Regions*; in this paper we focus on the 103 *Topic* categories. Note that in contrast to the results described in [27] we use the raw RCV1 corpus without any additional corrections, thus the obtained performance values should not be directly compared.

The documents were transformed into term vectors as described in Section III-B. According to [27], the documents were then split into a training set of 23,149 documents, and a test set consisting of 781,278 instances. This is a chronological split that selects all documents published within the first 12 days in the corpus as training documents, while retaining most of the complete year as test data. This asymmetric split resulted in the interesting fact that only a relatively small subset of all terms in the corpus occurred in the training set (Table VII).

Classification was then performed by training a linear SVM on the training set for each of the 103 available topics, i.e., for each topic we solved a binary classification problem. We used the SVM implementation from LIBLINEAR [28] with default parameters $C = 4$ and $eps = 0.1$ (solver `L2R_L2LOSS_SVC_DUAL`). The individual performance values obtained for each topic on the test set were then combined using micro- and macro-averages and compared for the same feature engineering scenarios described in Section III-C: i) using all available tokens, ii) using nouns and proper nouns tagged by the state-of-the-art maximum entropy based tagger provided by the OpenNLP library, and iii) using nouns and proper nouns tagged by our lightweight tagger based on Wiktionary.

Table VII shows the number of features extracted for each of these scenarios. Interestingly, about 90% to 95% of all tokens are nouns, indicating the rather factual nature of newswire articles. This further supports the hypothesis that in many real world scenarios, the restriction to nouns in feature engineering preserves most of the information contained in documents.

The runtimes of the Reuters classification algorithm in all three feature engineering scenarios are shown in Table VIII. All times were averaged over 10 runs on our test machine specified in Section III-A. It can be seen that in the vectorization stage the lightweight tagger significantly outperforms the heavy-weight OpenNLP tagger and is almost as fast as using no tagger at all (all tokens). The runtime of the SVM classification is only indirectly depending on the type of feature engineering used; since the other preprocessing steps (tokenization, stemming, stop word removal) and the number of data samples are the same for all three scenarios, its runtime mainly depends on the number of features, which is highest for the case where all tokens are used.

 ISBN: 978-1-61208-295-0

Table IX. COMPARISON OF THE CLASSIFICATION PERFORMANCE OBTAINED WITH DIFFERENT FEATURE ENGINEERING METHODS. SHOWN ARE MACRO- AND MICRO-AVERAGES OF PRECISION, RECALL, AND F1, RESPECTIVELY, FOR THE THREE DIFFERENT SCENARIOS.

	Macro-Averages			Micro-Averages		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
all tokens	0.446	0.692	0.519	0.721	0.852	0.781
OpenNLP	0.428	0.663	0.497	0.704	0.840	0.766
lightweight	0.418	0.657	0.491	0.691	0.835	0.756

Finally, Table IX shows the classification performance for the different feature engineering scenarios. Macro- and micro-averages are calculated over the individual performance values obtained by single binary classifiers on different topics. Micro-F1 values of about 0.75 to 0.8 are comparable to other studies on the RCV1 corpus [27], [29]. It can be seen that the performance is higher when all tokens are used compared to the cases where only nouns and proper nouns enter the feature space, which is expected since the higher the dimensionality of the input space, the more likely the linear classifier is able to find a good separating hyperplane. However, a comparison of only the noun-based preprocessing methods reveals that both the lightweight tagger and the OpenNLP tagger roughly achieve the same classification performance.

Thus, we can conclude that also for supervised classification tasks the significant reduction of runtime when using the lightweight tagger does not come at the cost of a decreased performance, even though the general restriction to nouns and proper nouns indirectly influences classification performance through the dimensionality of the feature space.

IV. APPLICATIONS & FUTURE WORK

Our lightweight tagger enables the processing of big data in tolerable amounts of time as compared to using more sophisticated PoS tagging models. As shown in Section III-A, our tagger has comparable recall for nouns and proper nouns compared to state-of-the-art PoS taggers. As outlined above, it lends itself well as a subsitute for more sophisticated PoS tagging models in various text machine learning tasks. However, the tagger as well as the parser and index for Wiktionary from which the tagger is built can be exploited for other tasks as well.

A recent hot topic in text mining literature is the extraction of facts from large (web) corpora [30]. Part-of-speech tagging plays a role in fact extraction, as it is often the basis for extracting patterns that could represent factual information. We envision a big data scenario, e.g., on the Common Crawl corpus [31], where our lightweight tagger can be used to extract a set of candidate patterns quickly. These candidate patterns can then be further refined by using a more accurate PoS tagger. Instead of having to tag the entire corpus accurately, we preselect a much more managable set of candidates to which we apply the costly, more precise PoS tagger. We plan on investigating this approach in future research.

The parser and index we devised for Wiktionary has other interesting application scenearios. Lemmatization [32] could greatly benefit from using Wiktionary as an information source. Wiktionary entries provide us with information on the lemma of words, together with all or most of their inflexions. This allows us to build a comprehensive dictionary that can be used to lemmatize known words. Developing a hybrid model of Wiktionary based lookup and statistical methods for unknown words is planned for future research.

V. CONCLUSION

Part-of-speech tagging is a crucial and time-consuming preprocessing step in many machine learning scenarios on natural language text. Our work tries to reduce the runtime of this step by approximating the performance of state-of-the-art PoS taggers, exploiting Wiktionary as the information source. We demonstrated that our approach can be used as a replacement for more precise but also more expensive PoS tagging models. The feature generation stage is considerably faster, while the quality of the machine learning results do not deteriorate. Furthermore, our approach is extendable to other languages without having to manually PoS tag large corpora of text. Instead, language collections from Wiktionary can be easily parsed and integrated. Our approach is especially well suited for big data scenarios, where short processing times directly translate into more experiments that can be carried out within a fixed time span. We described possible application scenarios and our planned future work, which includes fact extraction from large web corpora and lemmatization.

REFERENCES

[1] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.

[2] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

[3] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '94. New York, NY, USA: Springer-Verlag New York, Inc., 1994, pp. 232–241.

[4] C. Fox, "A stop list for general text," *SIGIR Forum*, vol. 24, no. 1-2, pp. 19–21, Sep. 1989.

[5] E. Brill, "A simple rule-based part of speech tagger," 1992.

[6] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, "A practical part-of-speech tagger," in *In Proceedings of the Third Conference on Applied Natural Language Processing*, 1992, pp. 133–140.

[7] A. Ratnaparkhi, "A maximum entropy model for Part-Of-speech tagging," in *Proceedings of the Empirical Methods in Natural Language Processing*, E. Brill and K. Church, Eds., 1996, pp. 133–142.

[8] P. V. S. Avinesh and G. Karthik, "Part-Of-speech tagging and chunking using conditional random fields and Transformation-Based learning," in *Proceedings of the IJCAI and the Workshop On Shallow Parsing for South Asian Languages (SPSAL)*, 2007, pp. 21–24.

[9] J. Gimenez and L. Marquez, "Svmtool: A general pos tagger generator based on support vector machines," in *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004, pp. 43–46.

[10] http://www.wiktionary.org/, [Online; accessed July 10, 2013].

[11] http://meta.wikimedia.org/wiki/Wiktionary, [Online; accessed July 10, 2013].

[12] http://en.wiktionary.org/wiki/Wiktionary:Criteria_for_inclusion, [Online; accessed July 10, 2013].

[13] https://www.knowminer.at/svn/opensource/components/ie/trunk/api/src/main/java/at/knowcenter/ie/postags/, [Online; accessed July 10, 2013].

[14] http://opennlp.apache.org/, [Online; accessed July 10, 2013].

 ISBN: 978-1-61208-295-0

[15] https://github.com/andyroberts/jTokenizer, [Online; accessed July 10, 2013].

[16] W. N. Francis and H. Kucera, "Brown corpus manual," Department of Linguistics, Brown University, Providence, Rhode Island, US, Tech. Rep., 1979.

[17] http://nltk.org/nltk_data/, [Online; accessed July 10, 2013].

[18] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: the penn treebank," *Comput. Linguist.*, vol. 19, no. 2, pp. 313–330, Jun. 1993.

[19] M. Porter, "An algorithm for suffix stripping," *Program: electronic library and information systems*, vol. 14, no. 3, pp. 130–137, 1980.

[20] M. F. Porter, "Snowball: A language for stemming algorithms," Published online, October 2001, accessed 11.03.2008, 15.00h.

[21] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters, *Text Processing with GATE (Version 6)*, 2011.

[22] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the Twelfth International Conference on Machine Learning*, 1995, pp. 331–339.

[23] http://qwone.com/~jason/20Newsgroups/, [Online; accessed July 10, 2013].

[24] S. Zhong, "Efficient online spherical k-means clustering," in *Proc. 2005 IEEE International Joint Conference on Neural Networks*, vol. 5, 2005, pp. 3180–3185.

[25] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, ser. SODA '07. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[26] C. D. Manning, P. Raghavan, and H. Schuetze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[27] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[28] R.-e. Fan, K.-w. Chang, C.-j. Hsieh, X.-r. Wang, and C.-j. Lin, "LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research*, vol. 9, no. 2008, pp. 1871–1874, 2012.

[29] R.-e. Fan and C.-j. Lin, "A Study on Threshold Selection for Multi-label Classification," Tech. Rep., 2007.

[30] M. Paşca, D. Lin, J. Bigham, A. Lifchits, and A. Jain, "Names and similarities on the web: fact extraction in the fast lane," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ser. ACL-44. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 809–816.

[31] http://commoncrawl.org/, [Online; accessed July 10, 2013].

[32] T. Korenius, J. Laurikkala, K. Järvelin, and M. Juhola, "Stemming and lemmatization in the clustering of finnish text documents," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, ser. CIKM '04. New York, NY, USA: ACM, 2004, pp. 625–633.

 ISBN: 978-1-61208-295-0

Efficient Optimization of Reinsurance Contracts using Discretized PBIL[1]

Omar Andres Carmona Cortes*, Andrew Rau-Chaplin†, Duane Wilson†, Ian Cook‡ and Jürgen Gaiser-Porter‡
*Informatics Academic Department
Instituto Federal de Educação, Ciência e Tecnologia do Maranhão
omar@ifma.edu.br, ocortes@dal.ca
†Risk Analytics Lab
Dalhousie University
arc@cs.dal.ca, dwilson@gmail.com
‡Global Analytics
Willis Group
cookiz@willis.com, gaiserporterj@willis.com

***Abstract*—Risk hedging strategies are at the heart of financial risk management. As with many financial institutions, insurance companies try to hedge their risk against potentially large losses, such as those associated with natural catastrophes. Much of this hedging is facilitated by engaging in risk transfer contracts with the global reinsurance market. Devising an effective hedging strategy depends on careful data analysis and optimization. In this paper, we study from the perspective of an insurance company a Reinsurance Contract Optimization problem in which we are given a reinsurance contract consisting of a fixed number of contractual layers and a simulated set of expected loss distributions (one per layer), plus a model of reinsurance market costs. Our task is to identify optimal combinations of placements such that for a given expected return the associated risk value is minimized. The solution to this high-dimensional multi-objective data analysis and optimization problem is a Pareto frontier that quantifies the best available trade-offs between expected risk and returns. Our approach to this reinsurance contract optimization problem is to adapt the evolutionary heuristic search method called Population Based Incremental Learning, or PBIL, to work with discretized solution spaces. Our multi-threaded Discretized PBIL method (or DiPBIL) is able to solve larger "real world" problem instances than previous methods. For example, problems with a 5% discretization and 7 or less contractual layers can be solved in less than 1h:20m, while previously infeasible problems that would have taken weeks or even months to run with as many as 15 layers can be solved in less than a day.**

***Keywords*—Financial Risk Management; Reinsurance Contract Optimization; Population Based Incremental Learning; Insurance and Reinsurance Analytics.**

I. INTRODUCTION

Risk hedging strategies are at the heart of financial risk management. As with many financial institutions, insurance companies try to hedge their risk against potentially large losses, such as those associated with natural catastrophes. Much of this hedging is facilitated by the global reinsurance market [1] (See Figure 1). Natural catastrophe reinsurers insure other insurance companies against the massive claims that can occur after events such as earthquakes, hurricanes, and floods. This transfer of risk is done in a manner similar to how a consumer cedes part of the risk associated with their private holdings by buying an insurance contract. This contract is defined in terms of a layer consisting of 1) a limit (i.e., the maximum payout), 2) a deductible or attachment (i.e., minimum loss triggering a claim), and 3) the share or placement (i.e., the percentage of losses in that layer that will be covered). Unlike the case of the consumer, the insurer has the ability to define complex multi-layered contracts and offer them to the reinsurance market. Doing so, it must carefully analyze the data it has on expected annual loss distributions and market reinsurance costs in order to identify contractual terms that maximize its expected reinsurance recoveries for each of a given set of risk tolerance values. In the insurance setting, typical risk measures include variance, Value at Risk (VaR) or a Tail-Value at Risk (TVaR) [1].

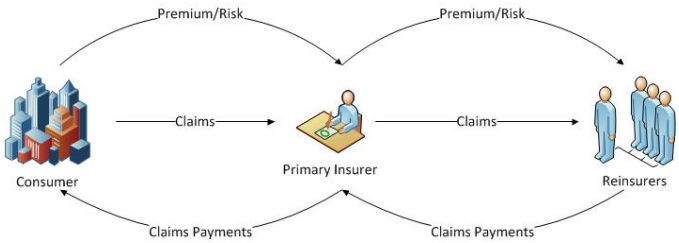


Figure 1. Risk and premium flows between consumers, primary insurers, and reinsurers

In this paper, we study from the perspective of an insurance company a *Reinsurance Contract Optimization problem*. Given a reinsurance contract consisting of a fixed number of layers and a simulated set of expected loss distributions (one per layer), plus a model of reinsurance costs, identifying optimal combinations of placements such that for a given expected return the associated risk value is minimized. The solution to this high-dimensional multi-objective optimization problem is

[1]This research was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Flagstone Re, Halifax, Canada under the Collaborative Research and Development Grant CRDPJ 412889-11. This project is also supported by the Science without Border program of CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brazil) and Instituto Federal de Educação, Ciência e Tecnologia do Maranhão

 ISBN: 978-1-61208-295-0

a Pareto frontier that quantifies the available trade-offs between expected risk and returns, as illustrated in Figure 2.

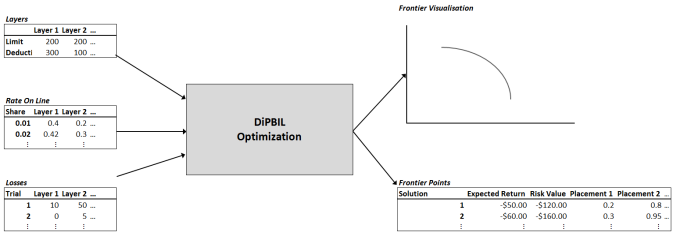


Figure 2. The studied reinsurance contract optimization problem: Inputs and Outputs

There are many heuristics methods that can be applied to optimization problems like this, such as Particle Swarm Optimization (PSO) [2], Differential Evolution (DE) [3], [4], Genetic Algorithms (GA) [5], Evolution Strategies (ES) [6] and Population-Based Incremental Learning (PBIL) [7]. Among these meta-heuristics we tried to use GA, nevertheless its performance was poor because demanded a lot of time to compute solutions considering our data set. Our approach is based on an adaptation of the evolutionary heuristic search method called Population Based Incremental Learning, or PBIL, which is a type of genetic algorithm where the genotype of an entire population is evolved rather than individual members and offers the following advantages. The algorithm is simpler than many standard genetic algorithms, works well in high dimensional spaces, and typically leads to equivalent or better results than standard GAs. Further, the algorithm was amenable to an important adaptation required by our problem, namely that solutions (i.e., placement values) must be discretized, typically in units of 10%, 5% or 1%, rather than being allowed to take on continuous values. Furthermore, PBIL also demands less computational power than the other methods since it is not based on genetic operators like crossover and selection.

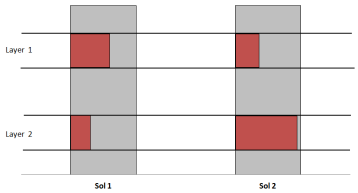
In the remainder of this paper, we first formally define our reinsurance contract optimization problem in Section 2. Then we describe a discretized PBIL method and how it can be applied to our problem in Section 3. In Section 4, we present a detailed performance analysis comparing our results to an enumeration method (both implemented in R) in terms of both speed and quality of result. Finally, we present the conclusions and future works in Section 5.

II. The Reinsurance Contract Optimization Problem

A. Reinsurance Business Basics

Insurance organizations, with the help of the global reinsurance market, look to hedge their risk against potentially large claims, or losses[1]. This transfer of risk is done in a manner similar to how a consumer cedes part of the risk associated with their private holdings.

Unlike the case of the consumer, whom is usually given options as to the type of insurance structures to choose from, the insurer has the ability to set its own structures and offers them to the reinsurance market. Involved in this process are decisions around the what type and the magnitude of financial



Solution	Layer	Limit	Deductible	Placement	Max Recovery	Premium
1	Layer 1	200	300	60%	120	$40
	Layer 2	200	100	20%	40	$40
2	Layer 1	200	300	30%	60	$25
	Layer 2	200	100	95%	190	$100

Figure 3. An example two layer reinsurance contract optimization problem with two sample solutions

structures, such as deductibles and limits, as well as the amount of risk the insurer wishes to maintain. The deductible describes the amount of loss that the insurer must incur before being able to claim a loss to the reinsurance contract, the limit describes the maximum amount in excess of the deductible that is claimable and the placement describes the percentage of the claimed loss that will be covered by the reinsurer.

Typically, companies try to hedge their risk placing multiple layers at once as illustrated in Figure 3. That is, they may have multiple sets of limit and deductible combinations. These different layers may also have differing placement amounts associated with them. At the same time, insurers are price takers in terms of the compensation paid to reinsurers for assuming risk. This compensation, or premium, depends on both the amount of risk associated with a layer and the placement amount of the layer. For this reason, it is important for insurers to choose placements when seeking to buy multiple layers. This optimal placement ensures that the insurer is able to maximize their returns on reinsurance contracts for potentially large future events.

In the remainder, we explore the use of optimization methods for finding optimal combinations of placements for multi-layer contracts from the prospective of a property-causality insurer. To simplify the problem description we focus on the primary contractual terms. Secondary terms such as the contractual costs associated with brokerage fee and contractual expenses, as well as provisions such as reinstatement premiums, are straightforward to add. As is typically done in reinsurance markets, contracts are assumed to be enforced for a one year period.

B. Reinsurance Costs

The basic cost of reinsurance to an insurer comes in the form of premium payments. As mention previously, the amount of premium paid for a layer can vary with the amount of the layer being placed in the market. In general, premiums are stated per unit of limit, also known as a *rate on line*. The cost of the reinsurance layer can then be expressed as follows:

$$p = \pi\mu(\pi, l, d) \times l \tag{1}$$

where p is the monetary value of the premium, μ is the rate on line, π is the placement, d is the deductible and l is the limit. For contracts with multiple layers, **(1)** can be generalized such that:

$$p = \vec{\mu}^T \mathbf{L} \vec{\pi} \tag{2}$$

where $\mathbf{L}$ is an $n \times n$ diagonal matrix of limits, $\vec{\mu}$ is a $n \times 1$ vector of rate on lines, $\vec{\pi}$ is a $n \times 1$ vector of placements, and n is the number of layers being placed. This matrix defines a model of expected reinsurance costs.

C. Reinsurance Recoveries

Losses affecting an insurer can be defined as a random variable X, such that:

$$X \sim f_X(x) \tag{3}$$

and f_X is some distribution that represents severity of X. These losses, once claimed, are subject to the financial terms associated with the contract they are being claimed against. Any one instance of X, x_i, then results in a claim of:

$$c_i = max\{0, min\{l, x_i - d\}\}\pi \tag{4}$$

where c_i is the value of the claim for i^{th} instance of X. Equation 4 can then be extended to contracts with multiple layers as follows:

$$c_i = \sum_{j=1}^{n} max\{0, min\{l_j, x_i - d_j\}\}\pi_j \tag{5}$$

where l_j, d_j and π_j are the limit, deductible and placement of the j^{th} layer respectively. In addition to this, many contracts allow for multiple claims in any given contractual year. The yearly contractual loss is then, assuming no financial terms that impose a maximum amount claimable, simply the sum of all individual claims in a given contractual year.

$$y_j = \sum_{i=1}^{n} c_{ij} \tag{6}$$

where y_j is the annual amount claimed for the j^{th} layer. The annual return for reinsurance contract is then defined as:

$$\begin{aligned} r &= \vec{y}^T \vec{\pi} - p \\ &= \vec{y}^T \vec{\pi} - \vec{\mu}^T \mathbf{L} \vec{\pi} \\ &= (\vec{y}^T - \vec{\mu}^T \mathbf{L}) \vec{\pi} \end{aligned} \tag{7}$$

where $\vec{y}$ is an $n \times 1$ vector of annual claims for each layer.

D. The Risk Value and Optimization Problem

Given a fixed number of layers and loss distributions the insurer is then faced with selecting an optimal combination of placements. As with most financial structures, the problem faced is selecting an optimal proportion, or placement, of each layer such for a given expected return on the contracts the associated risk is minimized. This is generally done by using a risk value such as a variance, Value at Risk (VaR) or a Tail-Value at risk (TVaR). The Tail-Value at Risk is also referred to as a conditional Value at Risk (CVaR) or a conditional tail expectation (CTE). Unlike, the traditional finance portfolio problem, in the insurance context a claim made, or loss, to the contract is income to the buyer of contract. This means, from the perspective of the insurer, they wish to maximize the amount claimable for a given risk value. In doing so they minimize amount of loss the insurer may face in a year.

Equation 7 can then be rewritten in matrix format such that:

$$\mathbf{R} = (\mathbf{Y} - \mathbf{ML})\vec{\pi} \tag{8}$$

where $\mathbf{R}$ is a $m \times 1$ vector of recoveries, $\mathbf{Y}$ is a $m \times n$ matrix of annual claims and $\mathbf{M}$ is a $m \times n$ matrix of rates on line. Since the same year is being simulated each row in matrix $\mathbf{M}$ is the same. This formulation leads to a optimization problem as follows:

$$\begin{array}{ll} maximize & VaR_\alpha(\mathbf{R}(\pi)) \\ s.t. & E(\mathbf{R}(\pi)) = a \end{array} \tag{9}$$

Given that the expected return a is not specified **(9)** can be rewritten to a Pareto Frontier problem, such that:

$$maximize \qquad VaR_\alpha(\mathbf{R}(\pi)) - qE(\mathbf{R}(\pi)) \tag{10}$$

where q is a risk tolerance factor greater than zero.

This problem can be approached in using numerous methods. Mistry et al. [14] use an enumeration approach by discretizing the search space for the each layer's placements. The discretization of the placements may be desirable for practical reasons (i.e., a placement with more than two decimal places may be invalid in negotiations) and the full enumeration method lends itself well to parallel computation. However, the computational time to evaluate all possible combinations increases exponentially as the number of layers and the resolution of the discretization increases. This renders the enumeration approach infeasible for many practically sized problems.

Mitschele et al. introduced the use of heuristic methods for addressing reinsurance optimization problems [13]. They show the power of two multi-objective evolutionary algorithms in finding non-dominated combinations, in comparison to the true non-dominated set of points. Mitschele et al., however, work exclusively in continuous space and focus on algorithms that change the limit and deductible aspects of a reinsurance contract, so there methods are not directly applicable here.

III. Discretized Population Based Incremental Learning

Population-Based Incremental Learning (PBIL) was first proposed by Baluja [7]. The algorithm's populations are encoded using binary vectors and an associated probability vector, which was then updated based on the best members of a population. Unlike other evolutionary algorithms, which transform the current population into new populations, a new population is generated at random using the updated probability vector on each generation. Baluja describes his algorithm as

 ISBN: 978-1-61208-295-0

a "combination of evolutionary optimization and hill-climbing" [7].

Since Baluja's work, extensions to the algorithm have been proposed for continuous and base-n represented search spaces [8], [9], [10]. The extension to continuous search spaces using histograms (PBIL$_H$) and real-code (RPBIL) suggest splitting the search space into intervals, each with their own probability [8], [10]. For multivariate cases, the probability vector is then substituted for a probability matrix, in such that each row or column of the matrix represents a probability vector for any given independent variable.

Algorithm 1 describes the discretized PBIL (DiPBIL) method used in this paper in terms of the following tunable parameters:

- N_G = Total number of generations or iterations
- n = Size of each population
- I = Number of Increments (i.e., the discretization)
- LR_2 = Learning Rate in base 2
- NLR_2 = Negative Learning Rate in base 2
- M_R = Mutation Rate
- M_S = Mutation Shift
- q = Number of best results to be used in updating

Algorithm 1: DiPBIL

Input: N_G, I, LR_2, NLR_2, M_R, M_S, n, q, function name(*fun*)
Output: $\mathbf{x}_G^{best}$, f_G^{best}
Initialization: P_{ij} = 1/I , LR_N, NLR_N, $\mathbf{x}_G^{best}$={}, $\mathbf{x}_i^{best}$={}

for i = *1* to N_G **do**
  Generate a population **X** of size n from P_{ij} ;
  Evaluate **f** = fun(**X**);
  Find $\mathbf{x}_G^{best}$ from the current and previous populations;
  Find $\mathbf{x}_i^{best}$ for top q-1 members of the current population;
  Update P_{ij} based on $\mathbf{x}_G^{best} \cup \mathbf{x}_i^{best}$ using LR_N and NLR_N;
end

While PBIL$_H$ and RPBIL support continuous search spaces, a similar method can be applied to a discretized search space. Here, we substitute the intervals for equidistant increments in the lower and upper bounds of the search space. This can also be related to a base-n representation, where each point in a given probability vector represents one number [9]. In the same spirit as PBIL$_H$ and RPBIL the probability matrix is initialized with all increments having equal probability and is updated after every generation with the best combinations member (see Algorithm 1). The updating of each vector in the matrix, however, is done using the base-n method, with an adjusted learning rate and updating function [9]. RPBIL suggests its own updating function which exponentially increases the probabilities as you move toward intervals that are closer to the best individuals [8]. The base-n updating method, however, has the side effect of slightly increasing the probability of increments further from the best individuals in the search space and may allow for a chance at more population diversity [9]. To ensure more population diversity from across generations, the probability matrix is updated with best member from previous generations as well as the top q members from the current generation. This modifies the updating equation slightly, such that:

$$p_{ij}^{NEW} = \sum_{k=1}^{q} p_{ij}^{OLD} \frac{LF_{ijk}}{q} \tag{11}$$

where LF_{ijk} is the i^{th} learning factor, as described in [9], for the k^{th} best result for the j^{th} variable.

IV. Performance Evaluation

This section presents the experimental evaluation of our reinsurance contract optimization technique. We first discuss our setup and methodology as well as the data sets used for the evaluation. We then present the performance results obtained in terms of quality of solutions and performance.

A. Experimental Setup and Methodology

We have implemented our discretized PBIL method using R and RStudio [11] and the doParallel parallelization package [12]. Our experimental platform consisted of a SunBlade server x6440, with four Quad-core AMD Opteron 8384 (2.7GHz) processors and 32 GB Ram, running Red Hat Enterprise Linux 4.8. The prototype code was written in R version x64 2.15.0 and doParallel version 1.0.1 with socket based connections.

All single threaded times were measured as wall clock times in seconds. All mutli-threaded times were measured as the wall clock time between the start of the first process and the termination of the last process. We will refer to the latter as *parallel wall clock time*.

For our experiments, we used anonymized industry data consisting of between 7 and 15 layers, with loss distributions represented by 50,000 trial loss sets, and a rate on line or reinsurance cost model defined in 10% increments with linear interpolation used between data points.

In all runs of DiPBIL, the following values were used for the fixed parameters: $LR_2 = 0.1$, $NLR_2 = 0.01$, $M_R = 0.02$, $M_S = 0.05$, and $q = 3$. These parameter settings were chosen based on recommendations from the literature followed by empirical testing. In experiments that varied the number of iterations, discretization, and/or population size the following values were used: $N_G = 500$, 1000 or 2000; $I = 10\%$ or 5%; and ($n = 100$, 200 or 400).

Our experiments proceeded in the following steps.

- Comparison of DiPBIL vs exact methods: Quality of results.
- Comparison of DiPBIL vs exact methods: Speed.
- Performance of multi-threaded DiPBIL.
- Pushing the envelope.

 ISBN: 978-1-61208-295-0

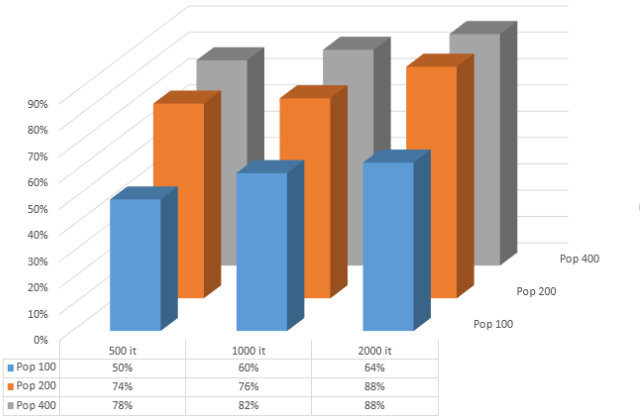


Figure 4. Percentage of time DiPBIL finds the same solution as the exact method for varying iterations and population size

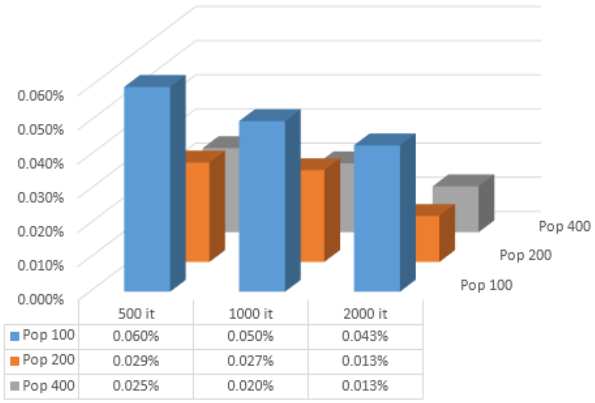


Figure 5. Average error when DiPBIL does not find the same solution as the exact method for varying iterations and population size

B. Comparison of DiPBIL vs exact methods: Quality of results.

Figure 4 shows the percentage of time DiPBIL finds exactly same solution as the exhaustive enumeration approach for varying iterations and population size on a problem with 7 layers and a 5% discretization. Each data point represents the average of 50 experiments. As expected, increasing iterations and population size increases the probability of finding exactly the same solution as the exhaustive enumeration approach, with $N_G = 2000$ iterations on a population size of $n = 200$ providing the best quality vs. time trade-off.

Figure 5 shows the average error for the cases where the exact solution is not found. We observe that the average error is always less than our predetermined error tolerance of 1%. The largest average error occurs, as expected, with the smallest population size and iteration count, but even in this case the average error is only 0.06% and with 2000 iterations on a population size of 200 the average error drops to just 0.013%. On other words, with 2000 iterations and population size of 200 the difference between the average of 50 runs and the optimum given by the enumeration method is 0.013%.

C. Comparison of DiPBIL vs exact methods: Speed.

The main reason behind the use of PBIL is the infeasible time to run the enumeration method, especially in large problems with many layers. Figure 6 illustrates the required time for different level of discretizations and number of layers, where shares 01, 05, 10 and 25 depict the discretization levels of 1%, 5%, 10% and 25%, respectively. For example, considering 7 layers and 5% of discretization, the enumeration method takes more than a week to get the answer.

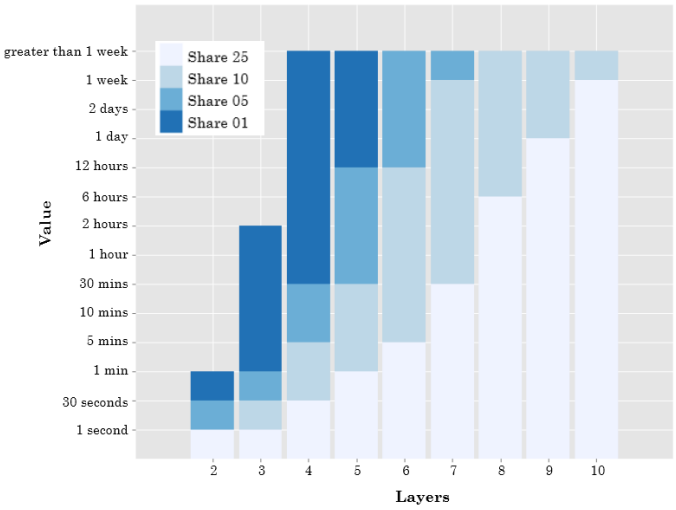


Figure 6. Required time for different number of layers and discretization

On the other hand, Figure 7 shows the time on a single core to compute a single point on efficient frontier for a 7 layer and 5% discretization problem. Note that the enumeration method takes over 6 days to run with a 5% discretization. The DiPBIL based approaches take between 100 and 1100 seconds to run depending on the population size and number of iterations. It is also important to note that DiPBIL is relatively insensitive to the discretization level, running in roughly the same time at 1% discretization, while reducing the discretization to 1% for the enumeration method will (by our estimation) increase the computing time to over a year. While this graph captures the relative performance of the methods all of the reported values based on a prototype R implementation could likely be reduce by a significant factor with a optimization implementation in C/C++.

D. Performance of multi-threaded DiPBIL.

Figure 8 shows the time and Figure 9 the corresponding speedup achieved by the multi-threaded version of DiPBIL, while computing a 32 point efficient frontier for a realistic 7 dimensional Treaty Optimization problem using a population size of 200 and 2000 iterations. Even with prototype code written in R a 75% speedup is achieved with results obtained in about 1h 20m. Early indications suggest that a tuned C/OpenMP implementation would be significantly faster.

E. Pushing the envelope.

In order to explore the limits of how large a treaty optimization problem we could effectively solve, we executed

 ISBN: 978-1-61208-295-0

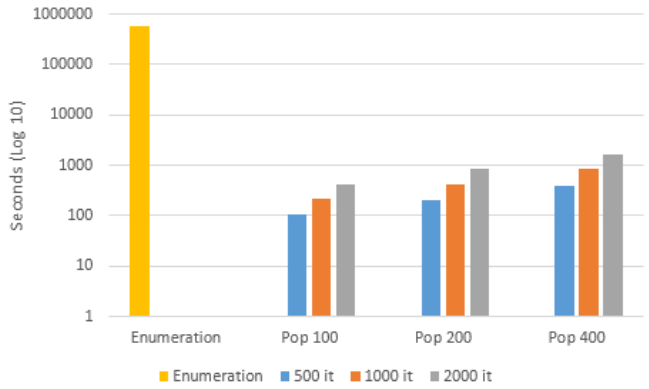


Figure 7. Comparing speed of DiPBIL for varying dimensionality

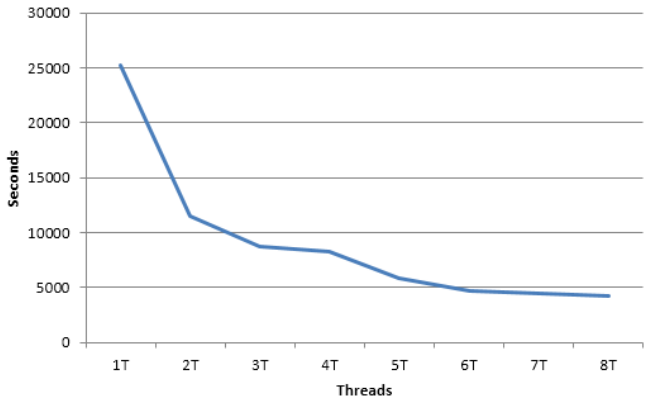


Figure 8. Performance of multi-threaded DiPBIL with increasing thread count

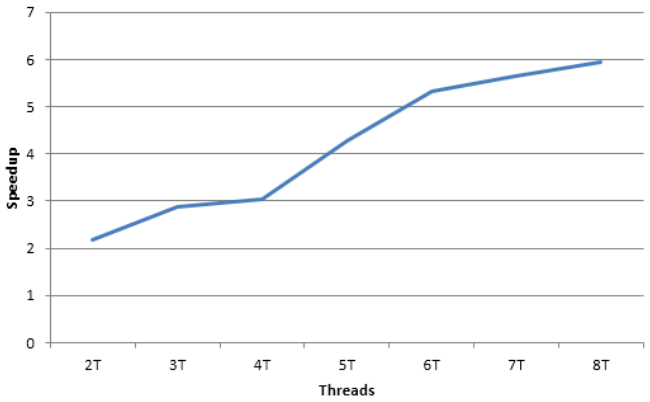


Figure 9. Speedup of multi-threaded DiPBIL with increasing thread count

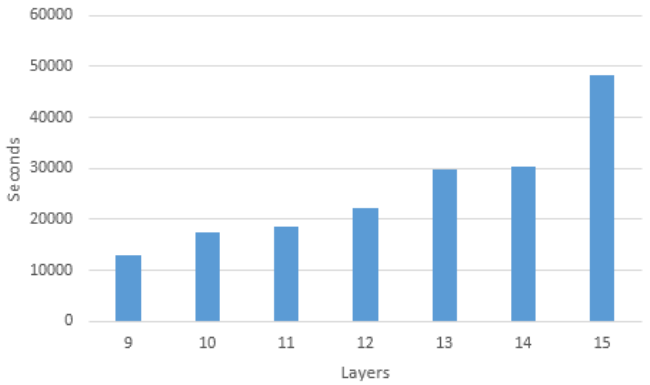


Figure 10. Time for multi-threaded DiPBIL to solve high dimensional treaty optimization problems with at fine level of discretization (5%)

multi-threaded DiPBIL on problems with between 9 and 15 dimensions and a 5% discretization with and measured the wall clock time. Figure 10 shows the results where each data point represents the average time to compute a 32 point frontier. We observed that DiPBIL was able to solve larger instances of the treaty optimization problem than in any previously reported implementation. Treaty optimization problems with as many as 9 layers were solvable in under 4 hours and massive problems involving 15 layers took significantly less than a day. This is a significant improvement since the run time for algorithms solving the treaty optimization problem typically grow exponentially in the number of layers.

V. Conclusion and Future Work

In this paper, we have studied a reinsurance contract optimization problem and shown that an approached based on a discretized adaptation of Population Based Incremental Learning generates high quality solutions in a time efficient manner. Either, Our multi-threaded DiPBIL method is able to solve "real world" problem instances with a 5% discretization and 7 or less layers in less than 1h:20m, and with up to 15 layers in less than a day. In contrast to exact enumeration methods, only treaty optimization problems with less than 7 layers are solvable in a day, while problems with 7 or more layers require a week or more of computation, or are simply infeasible.

This approach makes solving "real-world" problems with more than 8 layers and a 5% discretization feasible in a way it previously was not. Moreover, the differences between the DiPBIL and the enumeration method is not significant (lower than 0.06%) even when the lowest configuration (500 iterations and a population size equals to 200) is used.

In the future, we plan to extend our analysis to Differential Evolution and Particle Swarm Optimization. Also basic approach by examining alternative heuristic search strategies that might also support the required solution space discretization, evaluate the performance gains achievable with an optimized C/OpenMP implementation, and add constraints necessary to support secondary financial terms. Moreover, a multi-objective approach of DiPBIL is also in development.

References

[1] Cai, J., Tan, K. N., Weng, C. and Zhang, Y., Optimal reinsurance under VaR and CTE risk measures. *Insurance: Mathematics and Economics*, 43, 185-196, 2007.

 ISBN: 978-1-61208-295-0

[2] Kennedy, J. and Eberhart, R., Particle swarm optimization, Proc. of IEEE International Conference on Neural Networks , vol.4, pp. 1942-1948, 1995.

[3] Storn, R. and Price, K., Differential Evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, ftp.ICSI.Berkeley.edu/pub/techreports/1995/tr-95-012.ps.Z, 1995.

[4] Storn, R. and Price, K., Minimizing the real functions of the ICEC96 contest by differential evolution, Proc. of IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 1996.

[5] Michalewicz, Z., Genetic Algorithms + Data Structure = Evolution Programs, 3 ed, Springer, 1996.

[6] Yao, X., Liu, Y. and Lin, G., Evolutionary programming made faster, *IEEE Transactions on Evolutionary Computation*, v.3, n.2, pp. 82-102, 1999.

[7] Baluja, S., Population based incremental learning. *Technical Report*, Carnegie Mellon University, 1994

[8] Bureerat, S., Improved population-based incremental learning in continuous spaces. *Soft Computing in Industrial Applications*, 96, pp. 77-86, 2011

[9] Servais, M.P., Jager, G. and Greene, J.R., Function optimisation using multi-base population based incremental learning. *PRASA, Rhodes University*, 1997

[10] Yuan, B. and Gallagher, M., Playing in continuous spaces: Some analysis and extension of population-based incremental learning. *CEC2003, CA, USA*, pp. 443-450, 2003.

[11] http://www.rstudio.com/, Last Visit 20-May-2013.

[12] http://cran.r-project.org/web/packages/doParallel/index.html, Last Visit 20-May-2013.

[13] Mitschele, A., Oesterreicher, I. and Schlottmann, F. and Seese, D., Heuristic optimization of reinsurance programs and implications for reinsurance buyers. Operations Research Proceedings, pp. 287-292, 2006.

[14] Mistry, S. (n.d.), Gaiser-Porter, J. and McSharry, P. and Armour, T., *Parallel Computation of Reinsurance Models*. Unpublished Manuscript.

[15] http://www.ace-net.ca/wiki/Glooscap, Last Visit 20-May-2013.

 ISBN: 978-1-61208-295-0

Content-based Recommender System for Textual Documents Written in Croatian

Ivana Ćavar, Zvonko Kavran, Natalija Jolić
Neven Anđelović, Ivan Cvitić, Marko Gović
Faculty of transport and traffic sciences,
University of Zagreb
Zagreb, Croatia
ivana.cavar@fpz.hr; zvonko.kavran@fpz.hr, natalija.jolic@fpz.hr
neven1504@gmail.com; ivanfpz@gmail.com; marko.govic@gmail.com

***Abstract*—The paper describes a content-based recommender system that classifies textual documents written in Croatian. We describe how documents are pre-processed, including procedures of dimensionality reduction, selection of stop-words and creation of document-term matrix. For the text classification, a combination of *v*-fold cross validation and k - nearest neighbours (*k*NN) methods is used. This way, the 'optimal' value of *k* is firstly analyzed, and the results of *v*-fold cross validation are applied for the selection of value *k*. Results are given in the form of classification error analysis.**

Keywords-text mining; recommender system; k-nearest neighbour; content-based classification; document-term matrix.

I. INTRODUCTION

Search tools are one of the most used tools today. In the environment with different kind of sensors, available information, databases and Internet, the problem is not to find data, but to find useful information among available data and all that in the shortest possible time period and respectively with as little effort as possible along the way. This is one of the main reasons why recommender systems have been developed. They have the effect of guiding users in a personalized way to interesting objects in a large space of possible options. Every day examples for this include offering news articles to on-line newspaper readers, based on a prediction of reader interests, or offering customers of an on-line retailer suggestion about what they might like to buy, based on their past history of purchases and/or product searches.

This paper focuses on classification based recommender system developed for graduate students to aid their learning process by suggesting teaching materials based on a prediction of students' interests and past studying history. This problem is highlighted in multidisciplinary studding areas where students came with different levels of knowledge and different study backgrounds. The paper firstly introduces recommender systems and and a short literature review. After this, pre-processing of documents is described; this is followed by the description of text documents classification. The last section gives more details on results and, finally, we draw the conclusions.

II. RECOMMENDER SYSTEMS

Many areas have embraced recommender systems. There are many benefits from their applications; just some examples of that are the Google News' results with 38% more click-through due to recommendation, Netflix's rented 2/3 of movies based on recommendation, and 35% of Amazon's sales are from recommendation [1].

Recommendation systems use a number of different technologies and are basically classified into two broad groups [2]:

- Content-based systems examine properties of the items recommended;
- Collaborative filtering systems recommend items based on similarity measures between users and/or items. The items recommended to a user are those preferred by similar users.

Recommender systems for text documents written in natural language have become the subject of research for the past few decades. In literature [3] [4], examples of automatic classification of documents where web documents or articles are classified by Naïve Bayes algorithm [5] can be found. Other techniques are also applied for classification of textual documents. Description of library documents classification based on *k* - nearest neighbours algorithm can be found in [6]. Text mining-based recommendation systems assist customer decision making in online product customization, where customers describe their interests in textual format, and, based on captured customers' preferences, recommendations are generated [7]. Other examples can be found in [8] [9][10][11][12][13].

When modelling recommendation systems for text documents written in natural language it is important to carry out pre-processing procedures in order to provide good output results. Text documents considered in this recommendation system are multidisciplinary lecture materials written in Croatian.

 ISBN: 978-1-61208-295-0

III. PRE-PROCESSING OF TEXT DOCUMENTS

Firstly, it was necessary to represent text documents (strings) in a format suitable for text classification. For this purpose, a vector space model is used (Figure 1).

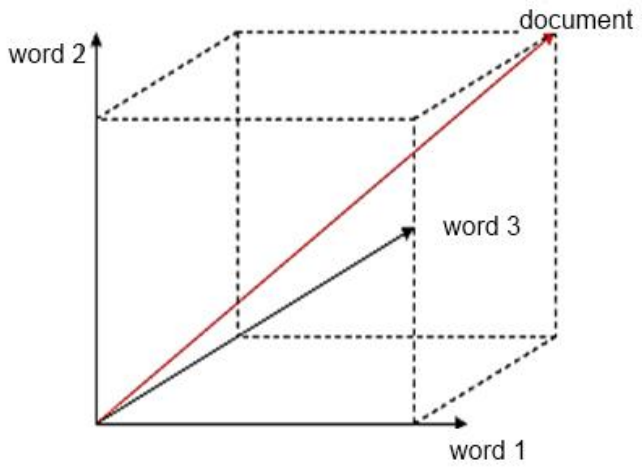


Figure 1. Text documents vector space

In this model, each text document is displayed as a vector of words. So, a document-term matrix or term-document matrix is created (Figure 2).

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & a_{ij} & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{matrix} w_1 \\ \dots \\ w_i \\ \dots \\ w_m \end{matrix}$$

$$d_1 \quad \dots \quad d_j \quad \dots \quad d_n$$

Figure 2. Document-term matrix

This is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms. For example, $A = (a_{ij})$, where a_{ij} is a weight of word in the document j. There are several ways of determining the weight a_{ij}. Let f_{ij} represent the frequency of words in the document j, N is the number of documents in the learning set, M is the number of different words, and n_i the total number of times a word appears in the learning set. The simplest approach for determining the weight is the binary weights approach, where a_{ij} is set to be *1* if the word appeared in the document; otherwise it is equal to *0*. Another simple method uses the frequency of occurrences of words in a document as a weight, i.e. $a_{ij} = f_{ij}$. The most popular way of determining the weight is Term Frequency - Inverse Document Frequency (TF- IDF) method of determining the weight where the weights are defined as:

$$a_{ij} = f_{ij} \times log\left(\frac{N}{n_i}\right) \quad (1)$$

If text documents with various lengths are considered, the adopted equation (1) looks like this (since for the matrix A the number of rows is determined by the number of different words in a text document):

$$a_{ij} = \frac{f_{ij}}{\sqrt{\sum_{i=1}^{M} f_{ij}^2}} \times log\left(\frac{N}{n_i}\right) \quad (2)$$

Given that there may be plenty of different words, including all the words in the language, plus the results of conjugation, and also, gender iterations (as in Croatian different words represent different genders), etc., it was necessary to determine keywords.

For this, the following steps were completed::

- Removing the stop – words;
- Tokenization;
- Lemmatization;
- Stemming;
- Synonyms;
- Group of words;
- Word cleansing.

A. *Stop – words*

Any group of words can be chosen as the stop - words for a given purpose. They can be defined as words that don't have a relevant meaning for the observed subject. Very often, the list of stop- words includes conjunctions, but in some other cases it depends on the document context. The list of stop - words varies depending on the morphological characteristics of the language so that the list for Croatian consists of approximately 2000 words while, for English, this number is approximately 600 words [14].

B. *Tokenization*

Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. Tokenization is a very important step in pre-processing documents written in morphologically rich languages like Croatian due to the fact that it allows dimensionality reduction of the input data [15].

C. *Lemmatization*

Lemmatization in linguistics is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. Lemmatization is a form of

 ISBN: 978-1-61208-295-0

morphological normalization or procedure that finds the linguistically correct canonical form of a word.

D. Stemming

Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form, generally, a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words are mapped to the same stem, even if this stem is not in itself a valid root. This type of morphological normalization is less accurate than lemmatization, because the root of the word does not necessarily have to be a meaningful expression [16].

E. Synonyms

Synonyms or synonymous words indicate that they have the same meaning. Their identification has a great impact in getting relevant results at the end of text analysis. For example, in Croatian 'ear of corn' has 47 synonyms *(ajdamak, bat, bataklјuša, bataljika, batučak, batuček, batuk, baturak, baturice, čepina, čokotinja, ćuka, kic, klas, klasina, klasinec, klasovina, klasovinje, kočanj, kocen, komaljika, komušina, kukuruzina, kumina, kureljica, kuruška, oklipak, oklasak, okoma, okomak, okomina, okrunica, orušek, otučak, paćika, patura, paturica, rucelj, rucl, rulina, šapurika, ščavina, šepurina, štruk, tekun, tulina, tulinek*). Similarly, as for the stop – words, the list is created for the synonyms.

F. Group of words

The term group of words represents a problem when a group of individual words, when written together, denote one meaning. For this purpose, we can use two approaches:

- Phrase list - combines word groups that represent common phrases in the language,
- Statistics - monitor the occurrence of two or more words together in the document. Group is defined as group of words if it appeared together more times than a predefined threshold. In order to increase the quality of the text that is the subject of analysis, such a group of words should be represented by one token.

G. Word cleansing

Word cleansing process is the last step in the pre-processing procedure. It includes removal of words that appear less frequently. Words that appear less than 1% of the time are usually the result of a typo error and the omission of such words reduces the noise of the document. The same is the case for the words that appear in the document more than 20% of the time.

As an input for document-term matrix creation 54 text documents written in Croatian were used. The used documents are in Portable Document Format (pdf) and selected form teaching materials written in Croatian. The main motivation for selecting these text documents was familiarity with content of this documents and their availability, as well as copy right issues. Based on the results, key words have been extracted for the word set representing each text document. This was done in two phases, manually and automatically. Based on the steps defined in this section, 984 keywords have been identified. These words were used for definition of document-term matrix.

IV. TEXT CLASSIFICATION

For text classification, weighted kNN method is used. When classifying a new document, kNN algorithm needs to determine the closest neighbours by calculating the distance vectors between documents [17]. Based on the *k* most similar neighbour class of considered document is decided. Similarity is determined by the Euclidean distance between vectors of documents or cosine value between two vectors of documents. Cosine value is defined as [18]:

$$sim(X, D_j) = \frac{\sum_{t_j \in (X \cap D_j)} x_i \times d_{ij}}{\left||X|\right|_2 \times \left||D_j|\right|_2} \quad (3)$$

where X is the vector of classifying document. D_j is the j -th document from the learning set, t_j is a word that exists in X and D_j, x is the weight of those words in the document D_j, $||X|| = \sqrt{x_1^2 + x_2^2 + x_3^2 + \ldots}$ is a normal vector X, and respectively $||D_j||$ is the normal vector D_j.

To determine the optimal size of neighbourhood, *v*-fold cross-validation method was applied. This means that, for a fixed value of *k*, we apply the kNN model to make predictions on the *v*th segment and to evaluate the error. The choice for this error is defined as the accuracy (the percentage of correctly classified cases). This process is then successively applied to all possible choices of *v*. At the end of the *v* folds (cycles), the computed errors are averaged to yield a measure of the stability of the model (how well the model predicts query points). The above steps are then repeated for various *k* and the value achieving the highest classification accuracy is then selected as the value for the *k*. Results of *v*-fold cross-validation in Statsoft Statistica [19] tool are shown in Figure 3.

 ISBN: 978-1-61208-295-0

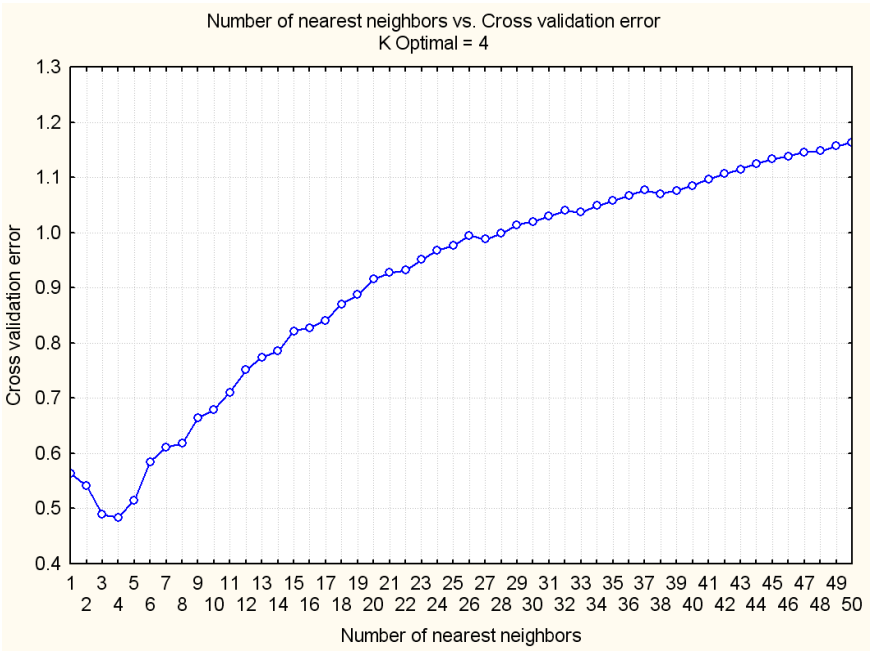


Figure 3. *v*-fold crossvalidation result

Based on the *v*-fold cross-validation results the value of the parameter *k* was set to be 4 (the lowest cross-validation error is 0.48429 %). For the *k* values that are higher than 4, continuous growth of classification errors is recorded. Figure 4 represents classification error where, in multidimensional space, classification error for 10 cases is represented by dots. As visible, most results are in the 'yellow' area, meaning that the value of the error was close to 0. Just one value is in 'red' area, meaning that error was in the interval between 3 - 5 %.

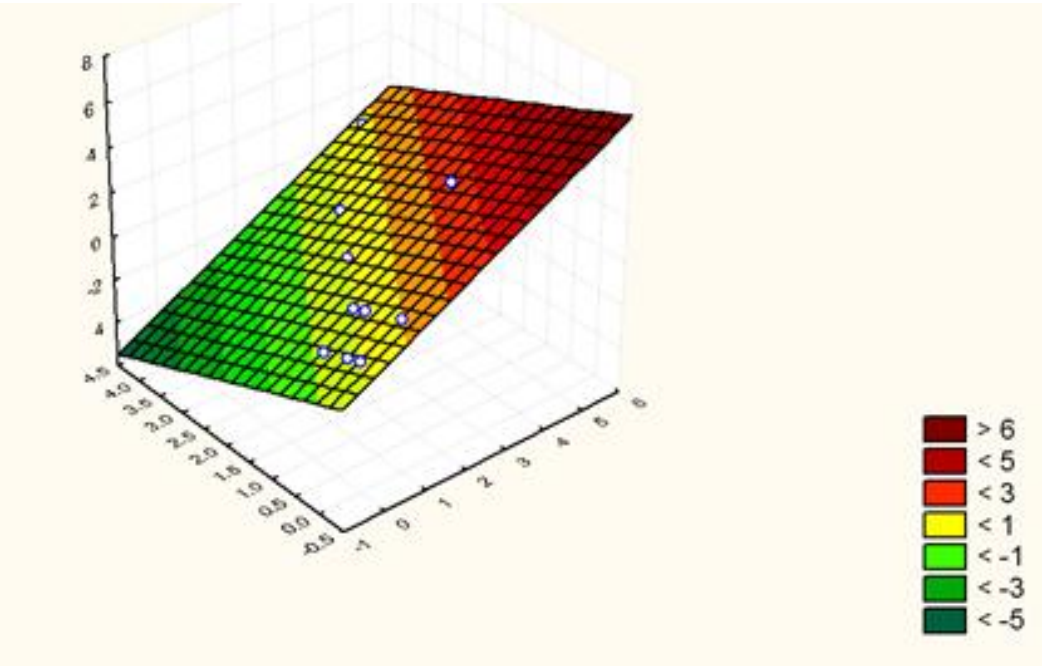


Figure 4. Classification error analysis.

 ISBN: 978-1-61208-295-0

The recommender system is modelled in such a way that on the basis of selected text document, output result lists semantically related teaching materials. This means that, when the student retrieves one lecture material, related teaching materials will be listed as suggestions for those who want to know more about the subject of the lecture.

V. CONCLUSION AND FUTURE WORK

Recently, we witness exponential increase in the amount of information being produced. Effective decision making based on such huge amounts of data can be achieved only if useful knowledge is extracted automatically from them.

The paper described an application of text mining techniques for extraction of useful knowledge from 54 text documents written in Croatian. This required text document pre-processing in order to define key words and form document-term matrix. Based on pre-processing, textual documents have been prepared for *v*-fold cross-validation method in order to define optimal size of document neighbourhood needed to classify it. For classification, kNN algorithm was used.

Applied process was used to classify lecture materials with aim to provide recommendations based on students' interests. This has proven to be useful in multidisciplinary areas such as traffic and transport engineering where students come with different studying backgrounds (e.g. information and communication technologies student needs to understand an multimodal urban traffic simulation teaching example to apply traffic control algorithm with public transport priority for signalised intersections). Application of developed recommender system allows students to consider lectures from different courses (for previous example, lectures on traffic control modelling course), regardless whether they have enrolled for his courses or have not.

In future research, it is planned to include more text documents as input and to create an interface that would allow students to access hyperlinks of suggested teaching materials. This is especially useful for students of multidisciplinary areas and those that have a wish to expand their knowledge.

REFERENCES

[1] Ò. Celma and P. Lamere, "If you like the beatles you might like...: a tutorial on music recommendation", ACM Multimedia, October 2008, pp. 1157–1158.

[2] A. Rajaraman and J. D. Ullman, "Mining of Massive Datasets", Cambridge University Press, Cambridge, UK, 2011

[3] Y. Wang, J. Hodges, B., Tang, B., "Classification of Web documents using a naive Bayes method", Dept. of Comput. Sci. & Eng., Mississippi State Univ. 2003, pp. 560–565.

[4] R.A Calvo., J. Lee and X. Li, "Managing content with automatic document classification", School of Electrical and Information Engineering, University of Sydney, Australia.2002.

[5] F. T. Hristea, "The Naïve Bayes Model for Unsupervised Word Sense Disambiguation: Aspects Concerning Feature Selection", SpringerBriefs in Statistics, 2013.

[6] J. Y. Pong, R. C. Kwok, R. Y. Lau, Y. Hao and P. C. Wong, "An unsupervised approach to automatic classification of scientific literature utilizing bibliographic metadata", Journal of Information Science, vol. 34, no. 2, 2008, pp. 213-230.

[7] A.R. Ittoo, Y. Zhang, J. Jiao, "A Text Mining-based Recommendation System for Customer Decision Making in Online Product Customization", The 3rd IEEE International Conference on Management of Innovation and Technology, Singapore, pp. 473-477

[8] S. Loh, F. Lorenzi, R. Saldana, D. Licthnow, "A tourism recommender system based on collaboration and text analysis", Information Technology & Tourism, vol. 6, no. 3, 2003, pp. 157-165

[9] S. Aciar, D. Zhang, S. J. Simoff, J. K. Debenham, "Informed Recommender: Basing Recommendations on Consumer Product Reviews". IEEE Intelligent Systems, 2007, 22(3): pp. 39-47

[10] R.M., Feitosa, S. Labidi, A.L. Silva dos Santos, N. Santos, "Social Recommendation in Location-Based Social Network Using Text Mining", 4th International Conference on Intelligent Systems Modelling & Simulation (ISMS), Bangkok 2013, pp. 67 – 72

[11] S. Venkatraman and S. J. Kamatkar. "Intelligent Information Retrieval and Recommender System Framework", International Journal of Future Computer and Communication, Vol. 2, No. 2, April 2013, pp. 85-89

[12] P. Lops, M. de Gemmis, G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends". Recommender Systems Handbook, 2011, pp.73-105

[13] M. J. Pazzani and D. Billsus, "Content-based recommendation systems. In The adaptive web", Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl (Eds.). Lecture Notes In Computer Science, 2007 Vol. 4321. Springer-Verlag, Berlin, Heidelberg pp. 325-341.

[14] C. Silva and B. Ribeiro, "The importance of stop word removal on recall values in text categorization", Proceedings of the International Joint Conference on Neural Networks, 2003, IEEE, pp. 1661-1666.

[15] M. Hassler. and G. Fliedl, "Text Preparation through Extended Tokenization, Data Mining VII: Data, Text and Web Mining and Their Business Applications". Volume 37. Edited by Zanasi, A and Brebbia, CA and Ebecken, NFF. WIT Press/Computational Mechanics Publications; 2006:pp. 13-21

[16] C. D. Manning, P. Raghavan, H. Schütze, "Introduction to Information Retrieval", Cambridge University Press. 2008.

[17] I. Aghayan, N. Noii, M. Metin Kunt, "Extended Traffic Crash Modelling through Precision and Response Time Using Fuzzy Clustering Algorithms Compared with Multi-layer Perceptron", PROMET - Traffic&Transportation, Vol 24, No 6, pp. 455-467

[18] N. Sandhya, Y.Sri Lalitha, A.Govardhan, "Analysis of Similarity Measures for Text Clustering, International Journal of Data Engineering", Vol 2, Issue 4, pp. 1-10

[19] http://www.statsoft.com/, May 2013

 ISBN: 978-1-61208-295-0

Finding Proteins Whose Expression Levels Depend on Bloodline in Wagyu

Takatoshi Fujiki
Graduate School of Systems Engineering,
Wakayama University, Japan
Email: s101044@sys.wakayama-u.ac.jp

Satoshi Sakaguchi
Graduate School of Systems Engineering,
Wakayama University, Japan
Email: s121016@sys.wakayama-u.ac.jp

Takuya Yoshihiro
Faculty of Systems Engineering,
Wakayama University, Japan
Email: tac@sys.wakayama-u.ac.jp

***Abstract*—Wagyu is known as beef brand with marbling character, in which the lineage of sires is significantly important to provide quality beef continuously. Sires of Wagyu have been improved through the dedicated efforts of inbreeding, to obtain excellent genetic ability to yield quality beef. In this decade, rapid growth of the technologies to analyze genes and proteins brings us a chance to improve the quality of beef using more direct and precise tools and knowledge. Tremendous amount of relations among genes, proteins, and traits have been clarified, and the knowledge can be potentially utilized to improve the quality of beef. However, there is scarcely a method that analyze bloodlines of sires and cattle to connect bloodline to genes and proteins. In this paper, we newly propose a method to treat bloodline of livestock animals on computers, and to find proteins whose expression levels have strongly related to the bloodline of cattle. With the proposed method, we firstly have a mean to know the relation between bloodline and proteins.**

***Keywords*—Beef Brand; Bloodline; Protein Expression.**

I. INTRODUCTION

Wagyu is Japanese native beef cattle known for marbling character, in which the linage of sires is significantly important to provide quality beef. Sires of Wagyu have been improved through the dedicated efforts of inbreeding, to obtain excellent genetic ability to yield quality beef. Currently, by using frozen sperms, quality beef cattle have been produced continuously from excellent genes of Wagyu sires. The lineage of sires, which guarantees the quality of Wagyu, is the precious genetic source that is essential to yield quality beef continuously.

For breeders of Wagyu cattle and sires, selecting sires (i.e., selecting sperms) for a new born cattle and a sire is one of the most important tasks, because the genetic character (hereafter, we call it the *lineage*) of a new born cattle and a sire is deeply related to the quality of beef yielded by them. Thus, traditionally, breeders utilize the values so called *breeding values*, which expresses the genetic ability of sires to produce the quality beef cattle, in selecting sires to use. In general, the breeding values are calculated using the BLUP (Best Linear Unbiased Prediction) method [1]. The BLUP method is a statistical prediction method to estimate the breeding values from the past results of beef grades of the sire's children, the bloodline information, and so on. Breeders of Wagyu have improved the genetic ability of sires for a long time by selecting good sires via breeding values to produce excellent descendants.

On the other hand, recently, many mechanisms of various life phenomena have been clarified due to the improvement of the technology to analyze genes and proteins of samples. For example, the techniques so-called microarray and 2D-electrophoresis enable us to measure expression levels of thousands of genes and proteins simultaneously [2]. With these high-throughput experimental methods, many specific mechanisms of creatures have been clarified so far. If the mechanism to yield quality beef is clarified, i.e., if the mechanism to connect both (i) from the bloodline to proteins, and (ii) from proteins to the beef quality, are clarified, a new and more efficient methodology to improve beef quality may be developed.

As for (ii), i.e., on the protein-protein interaction or the protein-phenotype relation, there are so many studies proposed so far. Bayesian networks [3] construct an interaction network model among proteins and phenotypes based on conditional probability. As other methods, we developed an algorithm to predict interactions among three proteins A, B and C, based on correlation coefficient [4], and conditional probability [5]. If we regard C as a phenotype, this method can be used to investigate the relation between proteins and phenotypes. However, there are few method that investigates (i), i.e., the relation between bloodline and proteins.

In this paper, we propose a new method that investigates the relation between bloodline and proteins; specifically, we try to find proteins whose expression levels are controlled by the lineage of beef cattle. As for these proteins, if there are two cattle that lineage is similar, then the expression levels are also similar, and otherwise the expression levels are not necessarily similar. By finding such proteins, we can determine a set of proteins in order to investigate the mechanism to improve the quality of beef, as well as we can specify the genes included in the lineage of sires that are deeply related to the expression levels of these proteins. To the best of our knowledge, this is the first study that tries to investigate the relation between the lineage and proteins.

This paper is organized as follows: In Section 2, we describe the protein whose expression levels depend on bloodlines, in addition to explain the input data of the proposed algorithm. In Section 3, we describe the proposed algorithm in detail. In Section 4, we present the method and the result of the evaluation. Finally, in Section 5, we conclude the paper.

II. PROTEINS WHOSE EXPRESSION LEVELS DEPEND ON BLOODLINE

A. Introducing Lineage Vectors to Represent Bloodlines

The genetic characters of Wagyu have been improved for a long time through dedicated efforts on inbreeding to generate

 ISBN: 978-1-61208-295-0

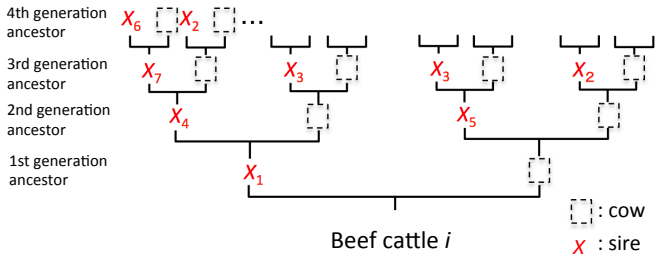


Figure 1. Family Tree of Cattle

excellent bloodlines of sires. All the ancestors of Wagyu cattle have excellent bloodlines, and the genetic character yields high-quality Wagyu beef. In general, the bloodline of each Wagyu cattle is recorded as a family tree that includes all ancestor sires over several generations. However, in order to treat bloodlines in computers, it is desirable to convert the family tree into a form with which we can easily treat it mathematically. So, we introduce a *lineage vector* of Wagyu cattle that expresses its genetic characters in a computable form.

Before the definition of lineage vectors, we first explain the records of family trees of Wagyu cattle. Figure 1 illustrates the family tree of one Wagyu cattle, where the root is the Wagyu cattle, and $X_1, X_2, \ldots, X_n$ are the ancestor sires of it. In general, cows are not included in the family tree because the effect of cows on genetic character is not strong; a sire can be the father of thousands of cattle whereas a cow can be the mother of no more than ten cattle. Note that a sire may appear more than twice in a family tree because Wagyu cattle as well as sires are generated from frozen sperms, and so the frozen sperms of an excellent sire tend to be used repeatedly even in different generations in the family tree.

Each Wagyu cattle has its family tree, and the lineage vector that is converted from the family tree. The line vector represents the ratio of genetic information inherited from each sire to the cattle. According to the genetic mechanism of inheritance, it is naturally assumed that the genetic information of cattle inherited from 1st generation (father) sire is 50%, and that from 2nd generation (grandfather) sire is 25%. Namely, the genetic information inherited from a k-th generation sire is $(\frac{1}{2})^k$. For instance, in Figure 1, 12.5% of the genetic information is inherited from the 3rd generation ancestors X_2, X_3, and X_7, and 6.25% from the 4th generation ancestors X_2 and X_6. If a sire appears more than twice in a family tree, the ratio of genetic information inherited is the sum of them, i.e., the genetic information inherited from X_2 is 12.5%+6.25%=18.75% in Figure 1.

Now, we define the lineage vector formally. Let $i(1 \leq i \leq b)$ be a cattle, $X_t(1 \leq t \leq n)$ be a sire, and $a_t^{(i)}(0 \leq a_t^{(i)} < 1)$ be the ratio of genetic information that cattle i is inherited from a sire X_t. Then, the lineage vector $B^{(i)}$ of cattle i is defined on the vector space where every possible sire has its corresponding basis, as follows:

$$B^{(i)} = \left(a_1^{(i)}, a_2^{(i)}, \cdots, a_t^{(i)}, \cdots, a_n^{(i)}\right) \ (1 \leq i \leq b) \qquad (1)$$

B. Expression Profiles of Proteins

Recent rapid growth of biological technology enabled us to analyze proteins included in a tissue of creatures with low cost in short time. There are several technologies that analyze proteins: one of major approaches is to obtain expression profiles, which analyzes the amount of each proteins included in a tissue, and the 2-dimensional electrophoresis is the representative technique to obtain expression profiles efficiently [2]. In this paper, we assume the protein expression profile of Wagyu cattle as the input data of the proposed algorithm.

We let $P_j(1 \leq j \leq m)$ be a protein, and the expression profile as the input of our algorithm consists of the expression levels $e_{P_j}^{(i)}$ for every proteins P_j and beef cattle i. We assume that the expression profile is normalized properly with some normalization methods.

C. Proteins Whose Expression Levels Depend on Bloodlines

In this paper, we try to find proteins that the expression levels depend on the bloodline of cattle, i.e., the expression levels are significantly related to the bloodline. Note that, if we choose a protein that is significantly related to the bloodline, the expression levels of two cattle of similar bloodlines will take similar values, and otherwise the two expression levels will have no relation. The problem that we try to solve is to measure the strength of this tendency between expression levels and bloodline for each protein in the expression profile.

We show an example in Figure 2 and Figure 3. Figure 2 is the case where the expression levels are related to the bloodline. Here, there are four vicinities of bloodlines and several samples (i.e., cattle) belong to them. The variance of each vicinity compared to that of all samples takes relatively small value. On the other hand, Figure 3 is the case where there is no relation between expression levels and the bloodline. The variance of every vicinity is almost the same as the variance of all samples.

Let us consider the problem more specifically using the lineage vectors of cattle. The lineage vectors belong to the *lineage space*, which is n-dimensional Euclidean space. If we suppose a point p in the lineage space, we can define the vicinity of p as the set of points within the Euclidean distance of ϵ. All we have to do is to examine every different coordinate in the lineage space, and for each of the coordinates, compute the variance of the expression levels of the samples included in the vicinity. If we compute the average of all the computed variances for each protein, the average indicates the strength of the relation between (the expression levels of) the protein and bloodline.

Note that there arises a problem of computation time because the lineage space has very large dimension n. In the next section, we will present the algorithm based on Gaussian processing [6] to cope with this problem.

III. The Proposed Method

A. Algorithm Design

As described in the previous section, we can retrieve the protein whose expression level is controlled by bloodline by examining the variance at every coordinate in the lineage

 ISBN: 978-1-61208-295-0

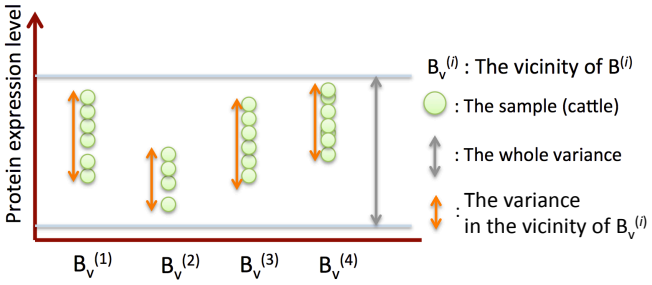


Figure 2. Protein Whose Expression Levels Depend on Bloodlines

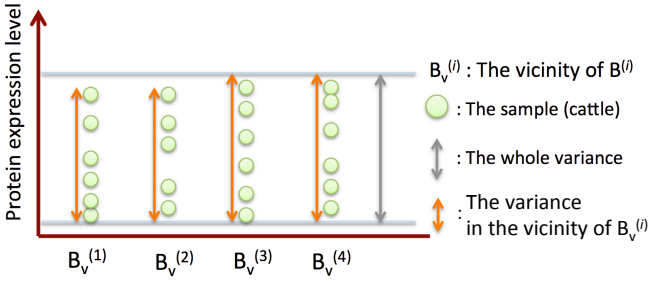


Figure 3. Protein Whose Expression Levels Don't Depend on Bloodlines

space. However, because the lineage space is continuous, to examine every possible coordinate in the lineage space is impossible. Moreover, because the dimension n of the lineage space is supposed as large as several hundred in a standard Wagyu dataset, the density of samples in the lineage space is very sparse even if the number of samples is several ten thousands. It is natural that the densities of samples at some coordinates are not large enough to guarantee statistical reliability of the computed variance values.

The main idea of the proposed algorithm is to partition the lineage space into many hypercubes (that we call *cells* hereafter) that have the same side length in every dimension, and we only examine the central coordinates of these cells. Note that the number of cells to be examined is tremendous because the lineage space has n dimensions. Thus, we only examine the cells to which at least one sample belongs to reduce the number of cells to examine. Furthermore, to guarantee the statistical reliability of computed variances, we calculate the variance of a cell only if the density of samples in the cell is larger than a threshold T.

Formally, the given parameter l, which represents the length of cell side, we define the coordinate of centers of cells in the n-dimensional lineage space as follows:

$$c = (c_1, c_2, \cdots, c_n), \tag{2}$$

where

$$c_t = ceiling\left(floor\left(\frac{a_t^{(i)}}{l/2}\right)/2\right) \cdot l \ (1 \leq t \leq n) \tag{3}$$

Then, the formal algorithm description is given in Table 1. In line 4, we compute the sample density at the center of each cell, and for each cells that densities are larger than T, we compute the variance of the cell in line 6. Finally in line 10, function compute_score() calculates the average of all the computed variances for each protein P_j as the *control score*, which represents the strength of the relation that the protein is controlled by bloodline. The lower the controlled score of a protein is, the stronger the expression levels of the protein are controlled by bloodline.

TABLE I. The Proposed Algorithm

```
1   foreach protein P_j (1 ≤ j ≤ m)
2     foreach cell c
3       if samples exist in c
4         Dens(c) ← compute_density()
5         if D_c ≥ T
6           Var(c, P_j) ← compute_variance()
7         end
8       end
9     end
10    Score(P_j) ← compute_score()
11  end
```

The functions to estimate the sample density (i.e., compute_density()), the variance of expression levels (i.e., compute_variance()), and the control score (i.e., compute_score()) are described in the following Sections III-B, III-C, and III-D, respectively.

B. Estimating Sample Density

In this section, we describe the function compute_density() that appears in line 4 of the proposed algorithm, which is the function to calculate the density of the center of a cell. Let $c = (c_1, c_2, \cdots, c_n)$ be the coordinates of the center point at which we want to compute the density. We apply the Kernel density estimation [7] to estimate the density, which is a widely used non-parametric density estimation method.

In our density estimation function, we first calculate the distance from the center c to each sample in the cell. Next, we estimate the sample density at the center c by accumulating the density according to the distance using the Kernel density function. Note that we use the Euclidean distance in this method.

Formally, the distance between the center c of the cell and the sample point of the beef cattle i are defined as

$$Dist(i,c) = \sqrt{\sum_{t=1}^{n}(a_t^{(i)} - c_t)^2} \ (1 \leq t \leq n) \tag{4}$$

Then, as the Kernel function $K(\frac{Dist(i,c)}{h})$, we use a general multi-dimensional Gaussian function, i.e.,

$$K\left(\frac{Dist(i,c)}{h}\right) = \frac{1}{(\sqrt{2\pi h^2})^n} exp\left(-\frac{1}{2h^2}(Dist(i,c))^2\right), \tag{5}$$

where h is a parameter that represents the bandwidth. As a result, the estimated density $Dens(c)$ of the center point c of the cell is given as follows:

$$Dens(c) = \frac{1}{b}\sum_{i=1}^{b} K\left(\frac{Dist(i,c)}{h}\right) \tag{6}$$

 ISBN: 978-1-61208-295-0

C. Estimating Variance of Expression Levels

In this section, we describe the function compute_variance() that appears in line 6 of the proposed algorithm, which computes the variance of the expression levels at the center of a cell.

We estimate the variance of expression levels using a Gaussian process. Namely, we calculate the weighted variance of expression levels of samples using a weight function where the weight is determined according to the distance between the center c and the point of samples. Formally, the estimated variance $Var(c, P_j)$ of expression levels for protein P_j at the coordinate c is represented as

$$Var(c, P_j) = \frac{\sum_{i=1}^{b} \left(\left(e_{P_j}^{(i)} - Avg_j(c) \right)^2 K\left(\frac{Dist(i,c)}{h} \right) \right)}{\sum_{i=1}^{b} K\left(\frac{Dist(i,c)}{h} \right)} \quad (1 \leq j \leq m), \quad (7)$$

where $e_{P_j}^{(i)}$ is the expression level of the protein P_j corresponding to the beef cattle i, and $Avg_j(c)$ is the average of the expression levels for protein P_j at c represented as follows:

$$Avg_j(c) = \frac{\sum_{i=1}^{b} \left(e_{P_j}^{(i)} K\left(\frac{Dist(i,c)}{h} \right) \right)}{\sum_{i=1}^{b} K\left(\frac{Dist(i,c)}{h} \right)} \quad (8)$$

D. Calculation of Control Score

We describe the process to calculate the controlled score of each proteins. The controlled score $Score(P_j)$ of the protein P_j is the average of the estimated value which the variance of the expression level at the central point $c^{(k)}$ $(1 \leq k \leq q)$ of the cell, and it is represented as follows:

$$Score(P_j) = \frac{\sum_{k=1}^{q} Var(c^{(k)}, P_j)}{q} \quad (9)$$

Note that the protein which we want to extract in this study is the protein that controlled score is low.

IV. EVALUATION

A. Model of Artificial Data Used in Evaluation

Because no protein controlled by the lineage of Wagyu is known, and so currently there is no real data that we can use to evaluate the proposed algorithm, we generate an artificial data set of proteins and the Wagyu lineages to evaluate the proposed algorithm. In generating an artificial data set, it is significantly important to construct a proper data model that reflects on the real property of the real phenomenon. Thus, we first propose a model of relation between genetic factors in lineage and expression levels of proteins.

In our model, we assume genetic factors that increase/decrease the expression levels of a protein, and they are inherited from ancestors to descendants in the genetic fashion. In general, currently, quantitative traits of creatures as well as protein expressions are regarded to be controlled by plural genetic factors (i.e., genetic polymorphism) [8]. Our assumption is based on this general agreement in the current state of the art.

First, we construct a realistic model of the lineage of Wagyu that properly explains the relation between beef cattle and sires. As described in Section II-A, several excellent sire lines exist in Wagyu as a result of long-time efforts of inbreeding to preserve and improve good genes that generate good quality beef. Thus, we model the sire lines as *sire-trees* that is rooted by a sire and its 10 generation ancestors are included in the tree, as illustrated in Figure 4. We regard that all the sires included in a sire-tree is distinct from others, and we prepare several sire-trees to express several major lines of sires.

Second, we generate and assign genetic factors to the sires in the sire-trees. In this study, we assume that the genetic factors that control the expression levels of a protein are not owned by a small portion of sires, rather broadly owned by sires with a certain probability, although the difference of distribution (i.e., sparse or dense) according to sire lines may be seen.

We designed the genetic factor model as follows: For each protein P_j, we generate r positive genetic factors $g_{j1}^p, g_{j2}^p, \ldots, g_{jr}^p$ and r negative genetic factors $g_{j1}^n, g_{j2}^n, \ldots, g_{jr}^n$, where these positive (resp. negative) genetic factors work to increase (resp. decrease) the expression levels of P_j. Here, note that, genetic factors are generally considered in pairwise fashion due to the pairwise nature of genomes. If we let 'A' be a genetic factor that works to increase/decrease expression levels, and let 'a' be a pairwise component that does not work on expression levels. Then, the genotype can be one of the three types 'AA,' 'Aa,' and 'aa.' We assign each of these genetic factors to every highest generation sire and cows with the genotype 'Aa,' and they are inherited to their descendants according to the law of genetic inheritance. Note that the genetic factors are inherited probabilistically to all the sires in the sire-tree, and the number of genetic factors on each sires is moderately distributed, which includes natural bias that coming from probability nature of inheritance.

Third, we generate beef cattle. The lineage of a beef cattle is uniquely determined if the sires to be ancestor are decided. Now, if we let 1st sire of cattle be its father, let 2nd sire be its mother's father, let 3rd sire be its mother's mother's father, and so on, only we have to do is to determine 1-5th sires for each cattle. So, for each beef cattle, we select 1-5th sires randomly from the sires of 5-10th generations in the generated sire-trees, as illustrated in Figure 5. This operation is done repeatedly for the number of required samples, and then the generated data set is regarded as the lineage data, which is the input of the proposed algorithm.

Finally, we generate protein expression profiles for each cattle. We assume that a protein expression level follow the normal distribution with the average μ and the standard deviation σ. Note that, although protein expression levels are generally regarded to follow log-scale distribution, there is a result in which protein expression levels follow the normal distribution [9]. For the data sets that follow log-scale distribution, we have only to apply logarithm to translate to the normal distribution. The expression level of a protein P_j in a sample i, i.e., $e_{P_j}^{(i)}$, is determined from the base normal distribution and the number of genetic factors corresponding to P_j in a sample (cattle) i. As for the function of genetic factors, we assume that if cattle has 'AA' or 'Aa' genotype as a result of the genetic probabilistic inheritance rule, the genetic factor works to increase/decrease

 ISBN: 978-1-61208-295-0

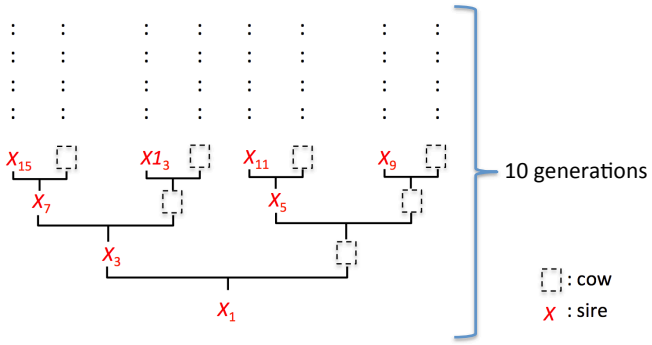


Figure 4. Sire Tree

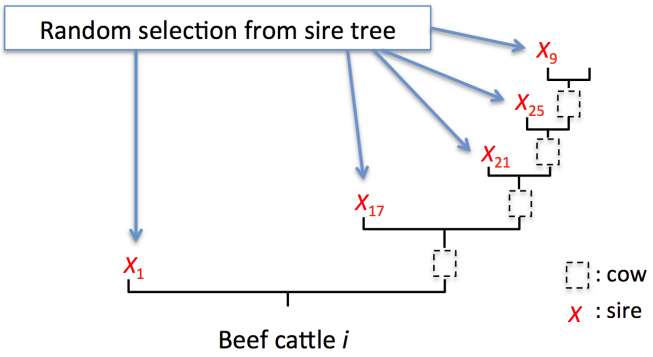


Figure 5. Generating Cattle

expression levels of the corresponding protein. We also assume that, if a genetic factor works, the average of the distribution μ is increased/decreased by a constant amount α. Namely, a set of expression levels corresponding to a protein and a set of samples (cattle) is generated probabilistically based on the normal distribution that average varies according to the number of the genetic factors that the sample have.

As above, we generate the artificial data according to the models of the lineage, the genetic factors, and the protein expression levels. A summary of the generation process of the artificial data is shown in Figure 6.

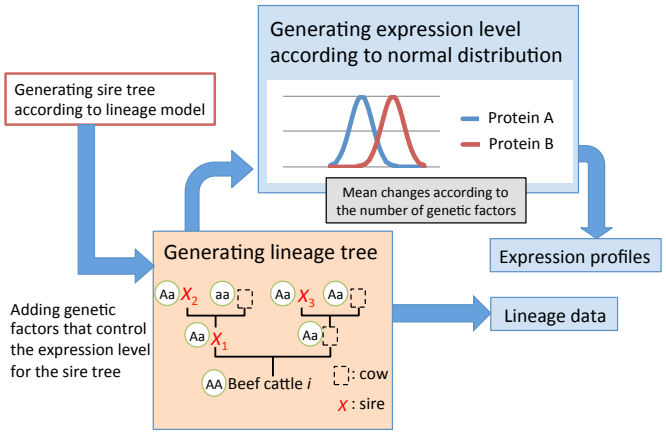


Figure 6. Process of Generating Artificial Data Set

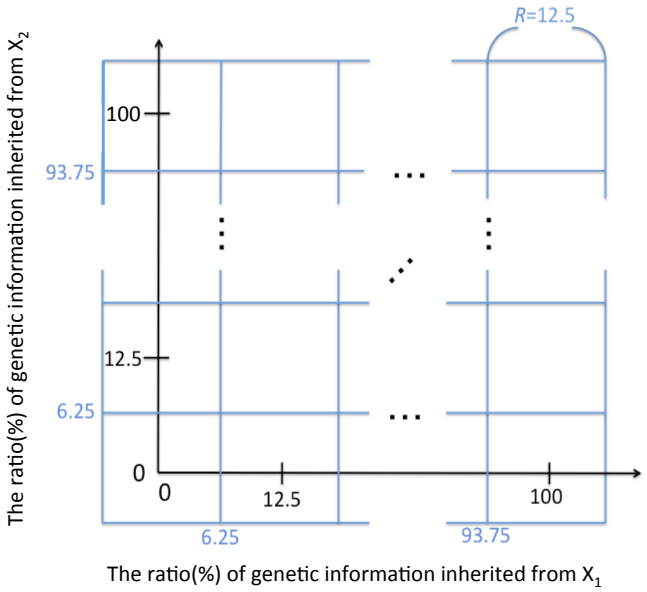


Figure 7. Dividing a Field into Cells

B. *Evaluation Method*

We generated a set of artificial data based on the models described in Section IV-A, and applied the proposed algorithm to it. We vary the number of genetic factors corresponding to each proteins; specifically, we choose the number randomly between 0 and 10. Then, if the control scores of a protein computed by the proposed algorithm is in relation to the number of genetic factors corresponding to the protein, it means that the proposed algorithm predicts the number of genetic factors, and further means that the control scores indicate how strong the expression levels of the protein depends on bloodlines.

In the following, we describe the conditions and parameters in the evaluation in detail. Based on the model described in Section IV-A, we generated a set of sire-trees, genetic factors, beef cattle, and expression profiles. In the lineage data, we have two sire-trees in which 10 generations of sires are included. The generated set of expression profiles includes 10,000 samples and 100 proteins, where the average and the standard deviation of the base normal distribution are $\mu = 0.5$ and $\sigma = 0.1$, respectively, and we let $\alpha = 1/4\sigma$ be the increment value of the expression level par genetic factor.

The cell width is set to $R = \frac{1}{2^3}$ and the centers of the cells are shifted so that the origin of the field (i.e., the point (0, 0)) is also the center of a cell, as shown in Figure 7. This is meant to have two cattle that has the same 1st, 2nd, and 3rd ancestors is likely to belong to the same cell in many cases. As for the parameters of the algorithm, we set the bandwidth of the Kernel function as $h = 0.21$ to cover cattle in a cell, and set the threshold of the density as $T = 0.7$ in consideration of the distribution of the density with the applied data set.

 ISBN: 978-1-61208-295-0

C. Results

The result of the evaluation is shown in Figure 8. Figure 8 is the scatter diagram where the horizontal axis represents the number of genetic factors and the vertical axis represents the control score, and the plotted points are the proteins. This result shows the strong correlation coefficient -0.838, which means that the proposed method succeeded to estimate the proteins that is controlled by the bloodline.

D. Discussion

By the simulation using an artificial data set, we demonstrated that the proposed method can predict proteins that are deeply related to bloodline. In this simulation, we assumed that each protein has the genetic factors in the DNA of sires, which control the expression level of the protein. This assumption would be widely acceptable because the genetic factors such as SNPs that control phenotypes or expression levels have been explored with tremendous efforts in the current scenes of biological studies.

The result of the simulation showed that the proposed method will work effectively to decide the target proteins to explore the system of living creatures; the protein that has many corresponding genetic factors would be in a position near genetic factors in the biological system, so that the direct interaction between genes and the protein will be found more likely than other proteins. As another practical usage of the proposed method, we suggest the possibility that the proposed method enables us to control important phenotypes more precisely and certainly by selecting better sires for a newborn cattle based on the knowledge of proteins. The proposed method would find proteins controllable by selecting sires, and the proteins that control an important protein will be found by other studies in the future.

There are several possibilities on how to use the knowledge obtained by the proposed method. To explore the practical use of the proposed method is an important task for the future.

V. Conclusion and Future Work

In this paper, we proposed a new method to predict the proteins whose expression levels depend on bloodline, from the lineage data and the protein expression profiles. Because bloodlines include dense genetic information, to connect the bloodline to proteins is valuable when we try to improve the quality of livestock animals through inbreeding. To the best of our knowledge, the proposed method is the first one that investigates the bloodline of brand cattle to connect to proteins.

To evaluate the proposed method, we designed a realistic data model of lineage, genetic factors, and expression profiles, and generated an artificial data. Through the evaluation using the artificial data, we confirmed that the proposed method can find the proteins whose expression levels are controlled by bloodline.

As future work to evaluate the effectiveness of this method firmly, it is desirable to have an evaluation using a real data set. However, the data set that includes an expression profile of proteins and the corresponding bloodline data is not currently available in public. Besides, even if such a data set is available, it is not possible by nature to grasp all the genetic factors that certainly effect on the expression level of a protein. Consequently, it is difficult to evaluate the accuracy of the proposed method exactly.

Considering this difficulty of real-data evaluation, one possible solution would be to demonstrate the effectiveness of the proposed method with some practical case studies. For example, to introduce the case in which the results of the proposed method contributed to a significant discovery or a practical methodology design, would contribute to prove the effectiveness of the proposed method. Although several difficulties are expected, to accumulate an achievement where this methodology worked effectively would be an important task for the future.

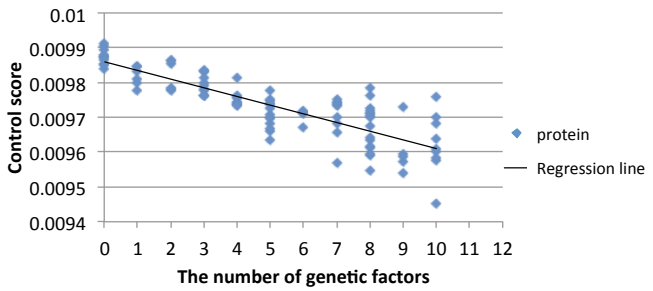


Figure 8. Correlation Between Control Score and Number of Genetic Factors

Acknowledgement

This work was partly supported by ”the Program for Promotion of Basic and Applied Researches for Innovations in Bio-oriented Industry” of NARO (National Agriculture and Food Research Organization), and ”the Program for Promotion of Stockbreeding” of JRA (Japan Racing Association).

References

[1] N. D. Cameron, “Selection Indices and Prediction of Genetic Merit in Animal Breeding,” CAB International, 1997.

[2] A. M. Campbell and L. J. Heyer, “Discovering Genomics, Proteomics and Bioinformatics,” Benjamin Cummings, 2006.

[3] N. Friedman, M. Linial, I. Nachman, and D. Pe’er, “Using Bayesian Networks to Analyze Expression Data,” Journal of Computational Biology **7**(3/4), pp. 601–620, 2000.

[4] E. Inoue, S. Murakami, T. Fujiki, T. Yoshihiro, A. Takemoto, H. Ikegami, K. Matsumoto, and M. Nakagawa, “Predicting Three-way Interactions of Proteins from Expression Profiles Based on Correlation Coefficient,” IPSJ Transactions on Bioinformatics, Vol. 5, pp. 34–43, 2012.

[5] T. Fujiki, E. Inoue, T. Yoshihiro, and M. Nakagawa, “Prediction of Combinatorial Protein-Protein Interaction from Expression Data Based on Conditional Probability,” Protein-Protein Interactions - Computational and Experimental Tools, InTech Web Press, pp. 131–146, 2012.

[6] C. E. Rasmussen and C. K. I. Williams, “Gaussian Processes for Machine Learning,” the MIT Press, 2006.

[7] R. O. Duda and P. E. Hart, “Pattern Classification and Scene Analysis,” John Wiley & Sons, Inc., 1973.

[8] T. A. Brown, “Genomes 2nd edition,” Garland Science Press, 2002.

[9] N. Balakrishnan and V. B. Nevzorov, “A Primer on Statistical Distributions,” John Wiley & Sons, Inc., 2004.

 ISBN: 978-1-61208-295-0

Discovering and Linking Financial Data on the Web

José Luis Sánchez-Cervantes
Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, Spain.
+34 91 624 5936
joseluis.s.cervantes@alumnos.uc3m.es

Gandhi S. Hernández-Chan
Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, Spain.
+34 91 624 5936
gandhi.hernandez@alumnos.uc3m.es

Mateusz Radzimski
Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, Spain.
+34 91 624 5936
mradzims@pa.uc3m.es

Juan Miguel Gómez-Berbís
Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, Spain.
+34 91 624 5936
juanmiguel.gomez@uc3m.es

Ángel García-Crespo
Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, Spain.
+34 91 624 5936
acrespo@ia.uc3m.es

***Abstract*—The constant publishing of large volumes of financial information by various business sector organizations through its financial statements is a fact that must be exploited by using semantic technologies. This paper describes the use of Linked Data to generate a financial dataset that is part of the ongoing work of the FLORA (Financial Linked Open Data Reasoning and Management for Web Science) project. Furthermore, we describe a process to discover other relevant links within the LOD cloud which can be related to FLORA dataset in order to provide a financial knowledgebase, which might be useful for: data analysis to support decision making, generating predictions or performing own financial discoveries to mention a few. However, the results of experiments performed, show that the coverage of the financial domain of public companies is rather small and ambiguous to link them to the FLORA dataset. Thus, is necessary involve more data as CIK (Central Index Key) or ticker symbol of the companies to objective of improve the results quality and implement techniques for disambiguation and manual verification.**

Keywords-Financial data; Semantics; Linked Open Data; FLORA; SILK framework.

I. INTRODUCTION

The current trend for companies to publish their financial statements under the Generally Accepted Accounting Principles (US-GAAP Financial model) and following the eXtensible Business Reporting Language (XBRL standard), increases the capacity for recovery and analysis of relevant data containing financial data semi-structured which is derived to facilitate the extending of financial datasets. Unlike datasets integrated by unstructured data and published on the Web, which represent a limitation by the high cost of transformation in order to be fit into existing analytic models and tools [1]. In our work, we have exploited the advantages of using semantic technologies [2], Linked Data (and Linked Open Data) [3] itself, which involves make the most of the best practices of sharing the data across the Web with great integration capabilities [4], basically we present a high level overview of an ongoing work in the FLORA project. FLORA fosters the transparent access to financial data through its dataset developed from extraction and triplification of the data contained in the financial statements of companies registered on U.S. Securities and Exchange Commission (SEC) [5] through fillings and forms stored on EDGAR System [6] furthermore, it allowing the easy combination of many information sources thus favoring the optimizing for data analysis. However, we consider important increasing the FLORA functionality through use of frameworks for discovering relationships between companies contained on FLORA dataset and data items within different Linked Data sources. This paper is structured in five sections, which are described briefly below. The introduction provides an overview of FLORA project, the second section corresponds to related work and it is divided into two subsections, the first subsection includes tools for discovery of links on the Web of data and second subsection describes some Linked Data projects. In section three, we describe the process for generation of financial dataset and discovery of related data items with companies stored on FLORA dataset. The fourth section, we describe the experiments for discovery of data elements in the LOD cloud and we present quantitative results obtained. Finally, we mention a conclusion and the future of our work.

II. RELATED WORK

In recent years, the systems for "triplification" and publication of datasets from multiple domains based on the Linked Data principles are becoming increasingly important. In this sense, there are several initiatives for the discovery of data items contained within the Linked Open Data cloud (LOD). We have classified these initiatives in two types, which are described briefly below:

A. Tools for discovery of links on the Web of data

Silk - Linking Framework [7], a toolkit for discovering and maintaining data links between Web data sources was presented in [8]. Silk is a tool that allows discovering

 ISBN: 978-1-61208-295-0

relationships between data items within different Linked Data sources. Data publishers can use Silk to set RDF (Resource Description Framework) [9] links from their data sources to other data sources on the Web. The presentation and evaluation of LIMES (Link Discovery Framework For Metric Spaces) [10] a novel time efficient approach for link discovery in metric spaces was described in [11]. The authors approach utilizes the mathematical characteristics of metric spaces to compute estimates of the similarity between instances. Thus, LIMES can reduce the number of comparisons needed during the mapping process by several orders of magnitude. In [12], LinQL framework was presented. It is a generic and extensible framework that works like an extension of SQL (Structured Query Language). This tool allows users to interleave declarative queries with interesting combinations of link discovery request. Its goal is to facilitate experimentation and help users find and combine the link discovery methods that will work best for their application domain. In [13], Silk Server was presented. It is an identity resolution component, which can be used within Linked Data application architectures to augment Web data with additional RDF links.

B. Linked Data applications

In [14], current efforts interlinking music-related datasets on the Web were addressed. The authors detail the application of an algorithm in two contexts: a) to link the Creative Commons music dataset to an editorial, and b) to link a personal music collection to corresponding Web identifiers. One of the most important features of this algorithm is that it was developed, implemented and practically it deployed to interlink different music-related datasets facing the overlapping problem. Faviki is an example of linked data application. It is a social bookmarking tool that lets users tag Web pages with semantic tags stemming from Wikipedia [15]. In [16], K-Search was presented. K-Search is an implementation of Hybrid Search (HS) another manifestation of linked data. It combines the flexibility of keyword-based retrieval with the ability to query and reason on metadata typical of semantic search systems. HS is defined as: i) the application of semantic search for the parts of the user queries where metadata is available and ii) the application of keyword-based search (KS) for the parts not covered by metadata however, KS is often affected by two main issues, ambiguity and synonymy. The LDIF-Linked Data Integration Framework [17] can be used with Linked Data applications to translate heterogeneous data from the Web of Linked data into a clean local target representation mapping language for translating data from various vocabularies that are used on the Web into a consistent, local target vocabulary. It includes an identity resolution component, which discovers URI (Uniform Resource Identifier) based on user-provides matching heuristic. The goal of Linking Open Drug Data (LODD) project presented in [18] is to facilitate the integration of large amounts of biomedical data from many different sources by bringing these data sources onto the Web of Linked Data. The biomedical datasets selected allow strong connections to existing Linked Data resources, while providing novel data of interest to pharmaceutical industry and patients. FLORA allows search of financial information contained in its data set including the calculation of several additional financial ratios that help users in finding information relevant to them [19]. However, the development of this project is iterative to obtain a continuous improvement at every stage of research and development whereby which we intend to exploit the FLORA possibilities. In the following section, stages of FLORA process are described briefly.

III. FLORA FINANCIAL DATASET

The transformation and generation process of FLORA dataset consists in investigate appropriate data sources and rich in financial information among which include: balance sheet, cash flow and income statements from companies. So far, we have generated a dataset that stores RDF triples generated from the extraction of 409.374 financial statements, which have been published in XBRL [20] format by different U.S. companies through the EDGAR system fillings [21].

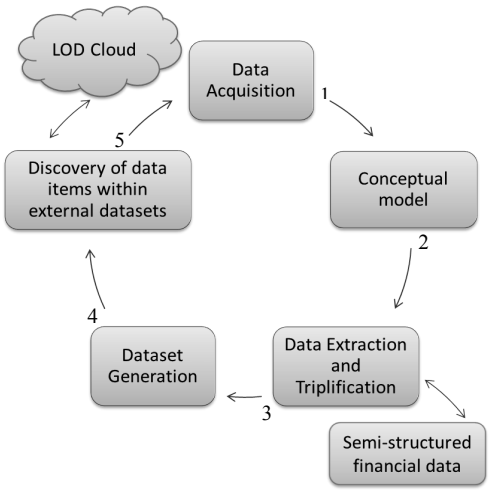


Figure 1. Stages of the FLORA process for generation of financial dataset and discovery of related data items

In Figure 1, each stage of the FLORA process has a defined function, which is briefly described below:

1. **Data Acquisition:** It is to research and get only the data sources of interest. The current focus of FLORA system is based on US-GAAP XBRL reports containing the Forms 10-Q, as published by the SEC EDGAR System. The published reports are crawled, downloaded and stored for further processing (Stage Data extraction and triplification). This stage needs to be repeated (at least quarterly) in order to retrieve newest reports and keep the system up-to-date.
2. **Conceptual model:** It is a meta-model that is the core of FLORA functionality because semantically represents the interaction between classes and subclasses that integrate it. This representation is

described in a high level of abstraction (see Figure 2) and includes an example of simplified taxonomy generated from published balance sheets under US-GAAP model including its general financial ratios, besides allowing the calculation of some additional ratios among which are: Total Asset Turnover, Non-Current Asset Turnover, Current Asset Turnover, Rotation of Rotation of Warehouse or Stocks, Working Capital, Cash Ratio, Debt Ratio and Ratio of Debt Quality.

3. **Data Extraction and Triplification:** this stage involves two simultaneous processes: the analysis (parsing) and extraction of data from financial statements made at the first stage and conversion to RDF triples extracted data in order to build the required dataset.
4. **Dataset generation:** this stage is closely related with the previous stage and consists in serialize the RDF triples to the semantic form. The dataset is following Linked Data principles and can be queried through SPARQL (Simple Protocol and RDF Query Language)-based queries.
5. **Discovery of data items within external datasets:** this is a stage that is currently underway and that we can consider the point to be discussed in this paper. As shown in Figure 2, the flow "*Discovery of related data items*" defines the search to find and create the links of all data stored in the LOD cloud that are related to companies registered in the FLORA dataset. Analyzing the state of the art (see Section 2) using frameworks like LIMES or Silk can be useful to achieve this goal.

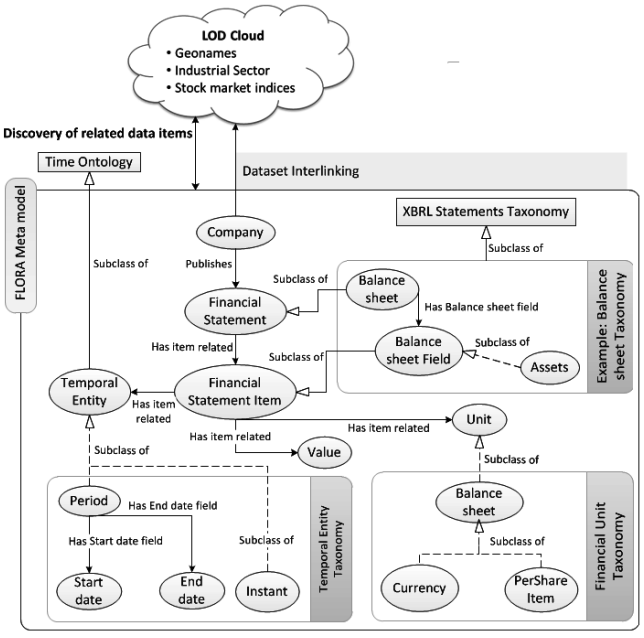


Figure 2. FLORA conceptual model

Discovering and linking additional data with data existing in FLORA dataset, will improve the search criteria and its reasoning capabilities. On other hand, through conceptual model we can identify the FLORA's behavior as well as detected and improve any data inconsistency with the possibility to correct or eliminate them.

IV. DISCOVERING DATA ITEMS IN THE LOD CLOUD

The volume of data stored in the FLORA system is increasing frequently to keep it updated. However, we find it interesting to investigate to what extent the LOD cloud covers the financial domain and what other data items may be linked with our dataset. To find this information, we have performed a set of experiments that allow linking external dataset with FLORA dataset through the SILK framework which provides the necessary mechanisms for discovering relationships between data items corresponding to FLORA dataset and different Linked Data sources within the LOD cloud.

As external datasets we have chosen DBpedia, Semantic XBRL and SEC triplified dataset. The latter two datasets are linking company concepts to DBpedia, but due to the outdated version (SEC triplified dataset discontinued as of 2010) and scarce mappings (Semantic XBRL) the only dataset is DBpedia.

The FLORA dataset that we have created is composed of 8915 public US companies obtained from SEC EDGAR fillings. In order to create mappings between concepts we considered comparing the following properties: (i) label, (ii) CIK (Central Index Key), (iii) ticker symbol. While the CIK and ticker symbol would clearly lead to the best results, most of the companies in DBpedia are lacking those data, leaving us with only label as a viable property for mappings.

To facilitate mappings, we performed 2 necessary steps: we created a list of stopwords for company names, including such terms as “inc”, “co”, “ltd” etc. and created a list of synonyms with stopwords filtered. In that example, a company “Apple Inc.” would have a synonym “Apple”.

For comparing strings, we used Levenshtein distance and a metric that measures the difference between two sequences. The Levenshtein distance between two strings is defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character.

After that, we launched experiments with FLORA dataset against DBpedia SPARQL endpoint, with following parameters:

- Experiment 1 presents a Levenshtein distance with value 0 and the use of Stopwords.
- Experiment 2 includes a Levenshtein distance with value 0 and no Stopwords (no synonyms, only the official company name).

 ISBN: 978-1-61208-295-0

- Experiment 3 contains a Levenshtein distance with value 1 and use of Stopwords.
- Experiment 4 involves a Levenshtein distance with value 1 and no use of Stopwords (as in Experiment 2).

The summary of the experiments performed and the results obtained are shown in Table 1.

TABLE 1. SUMMARY OF THE EXPERIMENTS PERFORMED AND THE RESULTS OBTAINED

Linking external dataset experiment			
Number of companies in FLORA dataset: 8915			
Experiment	***Levenshtein distance value***	***Stopwords***	***Discovered Links***
1	0	No	96
2	0	Yes	1031
3	1	No	437
4	1	Yes	3652

The results presented in this section are experiments for development of our work and can be considered as partial results for the stage of "Discovery of data items within external datasets" (see Section 3). While for a simple string comparison (experiment 1) we obtain quite a few mappings, this is what previously was observed by García and Gil [22]. However the striking difference is the increase of number of links created when a list of synonyms is generated. The distance between strings might include many false links, especially for short company names, but provides user with more possible alternatives in case of longer company names (with typically non-letter or a white space character difference).

V. CONCLUSION AND FUTURE WORK

This article presented a complex, unified process of transforming unstructured financial data into an interlinked, navigable knowledge base for financial information management and information discovery within the Web of data through the use of frameworks developed for this purpose. In the future work we aim at implementing LIMES framework, applying of inference using SPIN rules and identify and use other financial data sources to perform more complex experiments.

The experiment, however, shows that the coverage of the financial domain (in case of this paper: public companies) is rather small. The lack of data that could be used for univocally identify company concepts (such as CIK or ticker symbol) makes dataset interlinking still a difficult task requiring various techniques for disambiguation and manual verification.

In the future work, we are focusing on generation of rich gazetteer for each company in order to increase the number of possible matches based on the company name. Also, other string comparing metrics are considered; that could use other advanced features for comparing company names. After that, we will perform a manual evaluation of created links in order to assess the gazetteer and synonym list for company mappings.

ACKNOWLEDGMENTS

This work was supported by the Spanish Ministry of Industry, Tourism, and Commerce under the projects: FLORA (TIN2011-27405) and (SEMOSA TSI-020400-2011-51). It was also supported by The Spanish Ministry of Science and Innovation under the project GECALLIA (TSI-020100-2011-244). Additionally, this work was sponsored by The National Council of Science and Technology (CONACYT).

REFERENCES

[1] Ciccotello, C. S. and Wood, R. E. An investigation of the consistency of financial advice offered by web-based sources. *Financial Services Review*, 10, 1-4 (2001), 5-18. DOI=http://dx.doi.org/10.1016/S1057-0810(01)00078-6

[2] Allemang, D. and Hendler, J. *Semantic Web for the working ontologist: effective modeling in RDFS and OWL*. Morgan Kaufmann, 2011. ISBN-10: 0123859654.

[3] Bizer, C., Heath, T. and Berners-Lee, T. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5, 3 (2009), 1-22. DOI= 10.4018/jswis.2009081901

[4] O'Riain, S., Harth, A. and Curry, E. Linked data driven information systems as an enabler for integrating financial data. *Information Systems for Global Financial Markets*, Emerging Developments and Effects, (2011), 239-270. DOI=10.4018/978-1-61350-162-7.ch 010

[5] U.S. Securities and Exchange Commission, SEC. Retrieved June 16, 2013, from http://www.sec.gov/index.htm

[6] EDGAR System. Retrieved June 17, 2013, from http://www.sec.gov/edgar.shtml

[7] Silk - A Link Discovery Framework for the Web of Data. Retrieved June 17, 2013, from http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/

[8] Volz, J., Bizer, h., Gaedke, M. and Kobilarov, G. 2009. Silk–a link discovery framework for the web of data. *In Proceedings of the 2nd Linked Data on the Web Workshop LDOW2009*, (Madrid, Spain, April 20, 2009), 559-572.

[9] RDF, Resource Description Framework. Retrieved June 17, 2013, from http://www.w3.org/RDF/

[10] LIMES, Link Discovery Framework for Metric Spaces. Retrieved June 17, 2013, from http://aksw.org/Projects/LIMES.html

[11] Ngomo, A. N. and Auer, S. LIMES: a time-efficient approach for large-scale link discovery on the web of data. *In Proceedings of the Twenty-Second international joint conference on Artificial Intelligence* Volume Three. AAAI Press, 2011, 2312-2317.

[12] Hassanzadeh, O., Lim, L., Kementsietsidis, A., and Wang, M. A declarative framework for semantic link discovery over relational data. *In Proceedings of the 18th international*

 ISBN: 978-1-61208-295-0

conference on World Wide Web. (ACM New York, NY, USA, 2009), 1101–1102 DOI=10.1145/1526709.1526876

[13] Isele, R., Jentzsch, A., and Bizer, C. Silk Server-adding missing links while consuming linked data. *In 1st International Workshop on Consuming Linked Data (COLD 2010)*. (Shanghai, China, November 8, 2010)

[14] Raimond, Y., Sutton, C., and Sandler, M. Automatic Interlinking of Music Datasets on the Semantic Web. *In Proceedings of the 1st WorkShop about Linked Data on the Web LDOW2008*. (Beijing, China, April 22, 2008)

[15] Hausenblas, M. Exploiting Linked Data to Build Web Applications. *IEEE Internet Computing*, *13*(4), 68-73. DOI= http://doi.ieeecomputersociety.org/10.1109/MIC.2009.79

[16] Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V. and Petrelli, D. Hybrid search: Effectively combining keywords and semantic searches. *The Semantic Web: Research and Applications*, 5021 (2008), 554-568. DOI=10.1007/978-3-540-68234-9_41

[17] Achultz, A., Matteini, A., Isele, R., Bizer, C., and Becker, C. LDIF-Linked Data Integration Framework. *In 2nd International Workshop in Consuming Linked Data*. (Bonn, Germany, October 2011)

[18] Jentzsch, A., Cheung, K., Zhao, J., Samwald, M., Hassanzadeh, O., and Andersson, B. Linking Open Drug Data. *Presented at the Triplification Challenge of the International Conference on Semantic Systems*. (2009), 3-6.

[19] Radzimski, M., Sánchez-Cervantes, J. L., Rodríguez-González, A., Gómez-Berbís, J. M., and García-Crespo, Á. FLORA–Publishing Unstructured Financial Information in the Linked Open Data Cloud. *In International Workshop on Finance and Economics on the Semantic Web (FEOSW 2012)*, (Heraklion, Greece. May 27-28, 2012), 31.

[20] XBRL - Extensible Reporting Business Report Language. Retrieved June 19, 2013, from http://www.xbrl.org/

[21] EDGAR System Fillings. Retrieved June 20, 2013 from http://www.sec.gov/cgi-bin/browse-edgar?action=getcurrent

[22] García, R., and Gil, R. Linking XBRL financial data. *In D. Wood (Ed.), Linking enterprise data*. *Springer US*. (2010), 103-125. DOI=10.1007/978-1-4419-7665-9_6

 ISBN: 978-1-61208-295-0