Benelearn 2011

Proceedings

of the Twentieth

Belgian Dutch Conference on Machine Learning

The Hague, May 20 2011

Editors:
Peter van der Putten
Cor Veenman
Joaquin Vanschoren
Menno Israel
Hendrik Blockeel

Benelearn 2011

Proceedings

of the Twentieth

Belgian Dutch Conference on Machine Learning

The Hague, May 20 2011

Benelearn 2011. Proceedings of the Twentieth Belgian Dutch Conference on Machine Learning The Hague, May 20 2011.

http://www.benelearn2011.org

Table of Contents

Full papers

Extended Abstracts

Preface

Benelearn is the Annual Belgian Dutch Conference on Machine learning and the major forum for researchers in Belgium and the Netherlands to exchange ideas, present recent work, network and foster collaboration in machine learning, knowledge discovery in databases and data mining research and applications. This year is special, as it is the twentieth edition of Benelearn, making it one of the longest running regional machine learning conferences. It has been held on May 20, 2011 at the Netherlands Forensic Institute (NFI) in The Hague and has been organized by the Data Mining Group at LIACS, Leiden University and the Knowledge and Expertise Center for Intelligent Data Analysis (KECIDA), NFI.

In our view, Benelearn has always offered relevant and good quality content. For Benelearn 2011, we aimed at an increased focus on its role as a forum for the Belgian Dutch machine learning community. Therefore, we composed a 1-day program with a broad representation of the research in our community. The plenary program features a selection of the best full papers contributed, and invited talks covering broad subjects, as well as a poster session. This format is intended to make Benelearn more compact, more intense, and more relevant for a wider audience, including junior and senior researchers, from both academia and industry, covering research as well as applications of machine learning. At the time of writing this editorial Benelearn has been sold out with around 100 participants, with a good mix of backgrounds, and the accepted papers and abstracts provide a broad overview of research and applications.

With respect to the paper program, as mentioned, the intent is to provide a broad and rich overview of state of the art in research and applications. The extended abstract category is specifically aimed at this, by offering a broad scope for submissions. Extended abstracts may present current, recently published or future research, and can cover for instance position statements, offer a specific scientific or business problem to be solved by machine learning or describe a machine learning demo or art installation. In total, 27 extended abstract submissions have been accepted. To also drive excellence and differentiation we have selected a few distinguished papers. Only full papers presenting original, unpublished work covering research or a case application could qualify for this and 3 of these papers have been selected for plenary presentation, with the remainder of full papers and posters presented in an interactive poster session including flash talks.

In addition to the accepted papers and abstracts the program features invited talks on a mix of topics from researchers with varying backgrounds. The invited speakers are Aris Gionis, Yahoo! Labs (large graph mining), Kim Luyckx, University of Antwerp (machine learning and text analytics for user identification) and Arno Siebes, Utrecht University (minimum description length for pattern mining).

We would like to thank everyone who made Benelearn 2011 a success. This includes all the accepted paper and abstract authors, whose submissions form the backbone of the program; the invited speakers for their inspiring contributions; our main sponsor Pegasystems, our academic partners SIKS and BNAIC, and Miranda van Alem from NFI Academy for providing conference services. And last but not least, all the conference participants from academics and industry, who have made the conference a success.

The Benelearn 2011 Chairs

Organization

Benelearn is organized by the Data Mining Group at LIACS, Leiden University and the Knowledge and Expertise Center for Intelligent Data Analysis (KECIDA) at the Netherlands Forensic Institute (NFI).

Chairs:

- Peter van der Putten (LIACS)
- Hendrik Blockeel (LIACS)
- Joaquin Vanschoren (LIACS)
- Cor Veenman (Kecida, NFI)
- Menno Israel (Kecida, NFI)

Program committee:

Gianluca Bontempi	U.L. Bruxelles
Antal van den Bosch	Tilburg University
Joost Broekens	T.U. Delft
Walter Daelemans	University of Antwerp
Pierre Dupont	U.C. de Louvain
Damien Ernst	University of Liège
Ad Feelders	Universiteit Utrecht
Pierre Geurts	University of Liège
Bart Goethals	University of Antwerp
Tom Heskes	Radboud University Nijmegen
Edwin de Jong	Adapticon
Arno Knobbe	Universiteit Leiden
Joost Kok	Universiteit Leiden
Walter Kosters	Universiteit Leiden
Wojtek Kowalczyk	Vrije Universiteit Amsterdam
Ann Nowé	Vrije Universiteit Brussel
Martijn van Otterlo	K.U. Leuven
Mykola Pechenizkiy	T.U. Eindhoven
Mannes Poel	Universiteit Twente
Eric Postma	Tilburg University
Luc de Raedt	K.U. Leuven
Jan Ramon	K.U. Leuven
Yvan Saeys	Ghent University
Rodolphe Sepulchre	University of Liège
Evgueni Smirnov	Maastricht University
Maarten van Someren	University of Amsterdam
Johan Suykens	K.U. Leuven

Marten den Uyl	Sentient
Koen Vanhoof	Universiteit Hasselt
Celine Vens	K.U. Leuven
Katja Verbeeck	KaHo St-Lieven, Gent
Michel Verleysen	U.C. Louvain
Sicco Verwer	K.U. Leuven
Willem Waegeman	Gent University
Louis Wehenkel	University of Liège
Michiel van Wezel	Dudok Wonen
Menno van Zaanen	Tilburg University
Jakub Zavrel	Textkernel

Sponsors and partners:

Pegasystems	Main sponsor
SIKS	Academic partner
BNVKI	Academic partner
NFI Academy	Conference services

Invited Speakers

Text Analytics and Machine Learning for User Identification and Content Reliability

Kim Luyckx kim.luyckx@ua.ac.be

CLiPS Research Center, University of Antwerp

In this talk, I discuss applications of the stylometry methodology that are currently being investigated at the CLiPS research group. The goal of stylometry research is to identify the author and characteristics of the author, based on automatic analysis of the text, and more specifically, of the author's writing style. While these techniques typically have been developed for small closed-class problems, the methodology is challenged by the amount, variation (both linguistically and content-wise), authenticity, and quality of the textual information on the web.

After a brief description of these challenges, I will go into the various applications in user identification or profiling and the investigation of content reliability. For some applications, the mere identification of a content reliability or authenticity problem will be sufficient because there will be human correction and evaluation of the content. Intrinsic plagiarism detection – the task of identifying stylistic shifts in a text, a potential marker of plagiarism – is such an application since knowing the exact source of plagiarism is not an issue. However, when we consider the case where a user of a social networking site is leaking private material of another user, it is important to identify that person, even when fake user profiles have been made. Identifying the user's gender, age, personality can help identify the user. This has potential forensic applications, such as the identification of online groomers or the detection of breaches of personal privacy and dignity.

Bio:

Kim Luyckx (PhD in Computational Linguistics, Antwerp, 2010) was trained as a computational linguist at the University of Antwerp. She specialized in text categorization applications such as authorship attribution, authorship verification, and personality prediction. In 2010, she finished her PhD research on large-scale applications of authorship attribution. Currently she is a involved in a large-scale social media analysis project as a coordinator and postdoctoral researcher. Her main research interests are in large-scale text analytics, machine learning of language and digital humanities.

For more information, see http://www.clips.ua.ac.be/~kim/.

Efficient Tools
For Mining Large Graphs

Aris Gionis gionis@yahoo-inc.com

Yahoo! Labs, Barcelona

Graphs provide a general framework for modeling entities and their relationships, and they are routinely used to describe a wide variety of data such as the Internet, the Web, social networks, biological data, citation networks, and more. To deal with large graphs one needs not only to understand which graph features to mine for the application at hand, but also to develop efficient tools that cope with graphs having millions of nodes. In this talk we will review some recent work in this area. We will provide an overview of applications of graph analysis in query recommendation. We will then discuss algorithms for finding patterns of evolution, discovering communities, and forming teams of experts.

Bio:

Aristides Gionis is a senior research scientist in Yahoo! Research, Barcelona. He received his Ph.D from the Computer Science department of Stanford University in 2003 and until 2006 he has been a researcher at the Basic Research Unit of Helsinki Institute of Information Technology, Finland. His research interests include algorithms for data analysis and applications in the web domain.

For more information, see http://research.yahoo.com/Aris_Gionis

MDL for Pattern Mining

Arno Siebes Arno.Siebes@cs.uu.nl

Algorithmic Data Analysis Group, Information and Computing Sciences Utrecht University

Pattern mining is a popular topic in data mining. One problem is, however, the pattern explosion. That is, if one sets a high threshold one only gets well-known patterns. If one uses a low threshold one gets more results than data in the database. One way out of this predicament is to mine for sets of item sets. In our group we employ MDL for this problem. In this talk I will introduce two algorithms that aim to achieve this, viz., Krimp and Groei. We show that both algorithms return small sets of patterns that describe the data well.

For more information, see http://www.cs.uu.nl/staff/siebes.html

Distinguished full papers

Geopredict:
Exploring Models for Geographical Crime Forecasting

Bas Weitjens BWEITJENS@SENTIENT.NL
Gwennyn R. Kapitein GKAPITEIN@SENTIENT.NL
Marten den Uyl DENUYL@SMR.NL
Sentient Information Systems, Singel 160, 1015 AH Amsterdam, the Netherlands, www.sentient.nl

Keywords: geographical crime forecasting, parameter optimization, nearest neighbor, ARIMA

Abstract

Geographical forecasting can help the police plan more effectively by creating crime maps that show the likelihood of crime occurring at different places. In this paper different geographical crime forecasting models are compared. The models selected for the comparison are: two random walk models, an ARIMA model with a spatial extension and two fuzzy Nearest Neighbor (fNN) models. Before comparing these models the parameters for each of them are optimized. Random walk is optimized using field knowledge, the fNN models are optimized using an evolution strategy, and the ARIMA model is optimized using autocorrelation functions and gradient descent. The results of the comparisons between the different models show that ARIMA performs worst. The two fNN models perform consistently better than the random walk models. An fNN with extensive optimization performs somewhat better than with general optimization, but the differences in performance are small.

1. Introduction

Over the years the police have gathered a vast amount of crime data. Because the police have limited resources in manpower and time, these data are used to try to deploy the police forces more effectively; so called Information Led Policing. In the 90s geographic forecasts started with mapping the crime data on maps using upcoming geographic information systems (GIS). In 1998 the US National Institute of Justice (NIJ) awarded five grants to study how crime data could be better used for crime forecasting, for example by using data mining (Gorr & Harries 2003).

This paper focuses on geographical crime forecasting techniques, to forecast hotspots of crime. This kind of information can be used for tactical deployment of police forces. Various techniques are explored in order to compare their relative performance. In these comparisons, forecasts for varying situations are made, such as for different types of crime and for different prediction lengths. Different kinds of situations are investigated to see how widely applicable and robust the techniques are.

Because the forecasting techniques should perform well on different kinds of situations, the techniques had to be optimized. Because optimization takes considerable time, ways have been sought to do the optimization in advance instead of at run time. This led to optimization being a large part of the project.

1.1 Theory of Geographical Forecasting

All geographical crime forecasting techniques are based on some empirical theories. The repeat victimization theory states that one victim, or crime scene, can be the target of multiple crimes in a short time frame. Patterns matching this theory were found in multiple empirical studies, in different locations, and for different types of crime (Ericsson 1995, Bridgeman & Hobbs 1997).

Similar to repeat victimization, the near repeat victimization theory, introduced by Morgan (2000), states that targets near the previous victim have a higher chance of becoming a victim of crime in a short time frame after the first incident.

Johnson et al. (2007b) stated that the chance of being a victim of (near) repeat victimization decreases exponentially over time and space, which is a useful hypothesis when forecasting crime geographically for a given time.

Another theory used in crime forecasting is the theory of seasonality. According to this theory periodic fluctuations in crime rates occur, repeating themselves in seasonal intervals. Evidence for this theory was found in multiple empirical studies (e.g. Farrell and Pease, 1994, Block 1984). Seasonality can include weather attributes such as temperature, humidity, and many other factors, among others discussed by Field (1992) and by Rotton & Frey (1985). Besides seasonal effects, other temporal effects could also be present, such as the effect of different days of the week. All these factors could be of interest when forecasting crime.

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).

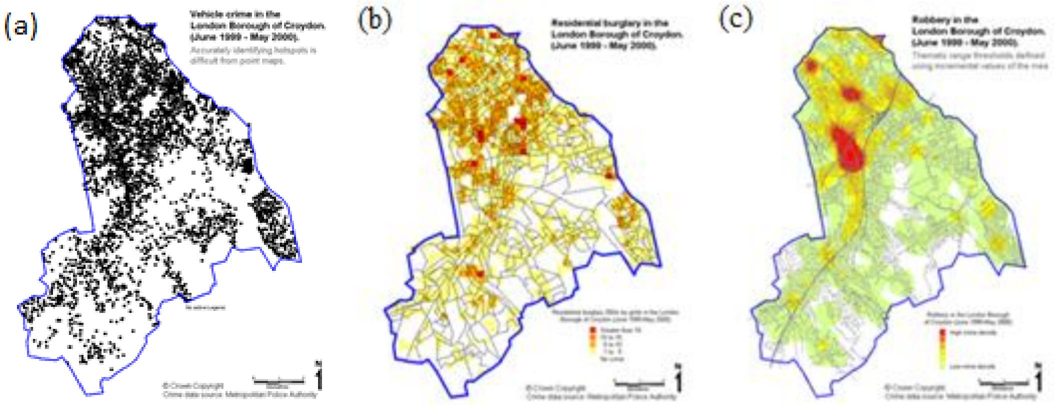

(a)
Vehicle crime in the London Borough of Croydon. (June 1999 - May 2000).
Accurately identifying hotspots is difficult from point maps.
(b)
Residential burglary in the London Borough of Croydon. (June 1999 - May 2000).
(c)
Robbery in the London Borough of Croydon. (June 1999 - May 2000).
Thematic range thresholds defined using incremental values of the mea

Figure 1. Different kind of crime data visualizations; (a) a point map, (b) a grid thematic mapping, (c) a smooth surface map. (Images from http://webarchive.nationalarchives.gov.uk/20110220105210/crimereduction.homeoffice.gov.uk/toolkits/fa00.htm)

2. Methods

2.1 Crime Forecasting Techniques

Using a GIS, crime data can be visualized in multiple ways: e.g. by plotting each incident as a dot on the map (as in Figure 1a), by coloring regions of the map based on the number of incidents that happened there (as in Figure 1b), or by smoothing the incidents with a Gaussian distribution to get a smooth surface map (as in Figure 1c).

Crime forecasting techniques use a GIS to visualize data that are selected and processed in a smart way. Many different techniques exist for making this “smart” selection of data. Some of these techniques are very simple and just select data for a specific moment in the past; so called retrospective techniques. Among these techniques is the random walk technique.

As mentioned, the risk of (near) repeat victimization decreases over time. So when an incident happened a day ago, the chance is higher that another incident will happen at (approximately) the same location than when an incident happened a month ago. This kind of rules can be incorporated into more advanced (prospective) techniques to make smarter selections. Also other factors, such as day of the week and time of the day, could have some impact, and the use of temporal patterns can be incorporated into forecasting techniques.

Three techniques were selected for analysis. Random walk was selected as a benchmark technique. This technique was selected because it is simple and it is used by the police in the Netherlands. The second selected technique was ARIMA with a spatial extension, because this technique takes a totally different approach than other techniques in literature, and it was therefore interesting to find out how well it would perform. The final technique selected was the fuzzy Nearest Neighbor (fNN) technique. This technique was selected because this is currently used by the company at which this research was done[1].

2.1.1 Random Walk

The random walk model visualizes data from a specific time interval in the past to approximate what will happen in the future. In the experiments for this paper a crime map was made by creating a smooth surface map of the past data.

The only parameter of this model is from which moment in the past the crime map should be created. Therefore this interval should be chosen very carefully, using knowledge of the situation to forecast for. The moment in the past should resemble the situation to forecast for as much as possible. The underlying assumption for this is that crime will happen at similar locations when the circumstances are similar.

2.1.2 ARIMA with a Spatial Extension

ARIMA (Box 1976) is the leading algorithm to forecast time series. To create forecasts it evaluates data from the past and tries to find temporal patterns in these data. To do this autoregressive (AR) and moving average (MA) models are combined, and possibly differencing (I) is used. This combination of models will look at known data of time intervals in the past to make an estimate for the next time interval. This approach is consistent with the seasonality theory mentioned earlier.

While ARIMA was designed for time series, it was used as a crime forecasting model by Pieterse (2007) in his thesis. In order to create geographical forecasts, the surface to forecast for is split into parts using a grid. For each cell in that grid all incidents within time intervals are summed up to form a single time series per cell. This way the spatial factor of the data for that cell is accounted for, and ARIMA is used to estimate the amount of crime in each cell separately for the next time interval.

For the ARIMA model to find patterns in the data it needs to be trained for each forecast on past crime data within each cell.

2.1.3 Fuzzy Nearest Neighbor

The fuzzy Nearest Neighbor (fNN) technique is an extension of the k Nearest Neighbor (kNN) technique (Cover & Hart 1967). In both techniques all data points (in this case incidents) are plotted in N-dimensional space, where N is equal to the number of attributes in

[1] Used in the product DataDetective of the company Sentient Information Systems

the data. The crime data has many different attributes that can be used:

- the date/time an incident happened, to calculate the recency of an incident
- other information that can be derived from the date/time, such as the month, day of the week, time of the day,
- the type of crime,
- other things that are known, such as the weather at the time of the incident.

When trying to forecast for a certain situation, that situation is also plotted in N-dimensional space using data known in advance, such as the date the forecast is made for, the type of crime to forecast for, and weather forecasts.

The standard kNN algorithm selects the k data points with the smallest distance to the plotted situation to approximate that situation, where the number for k is set in advance. Each of these k data points is considered equally important. With fNN also a, typically larger, number of data points with the smallest distance to a situation is selected. However, unlike kNN, the importance of each data point is a weighed function of the distance to the situation, hence the name fuzzy nearest neighbor.

With kNN usually the Euclidian distance is used, without explicit weighting attributes. For the fNN a weighted distance measure was employed. Weights are assigned to each of the attributes, and used to calculate distances in the corresponding dimensions. The weights can be optimized for different situations, because in different situations other attributes might be more important.

From the k incidents selected by the fNN, a continuous surface map is created, such as the one shown in Figure 2. The most similar past incidents will have a larger impact on this map than those with higher distances.

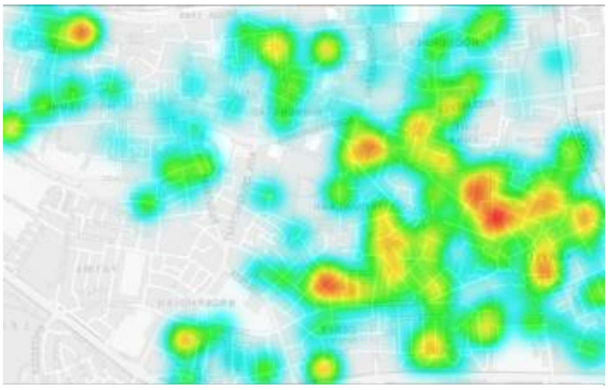


Figure 2. An example prediction map created for burglary incidents in January.

2.2 Performance measurements

To be able to optimize and compare the techniques, a performance measurement had to be selected. Because a forecast map cannot be directly compared to a set of real incidents some adaptations had to be made.

After considering different measurements, in the end two measurements were selected. The first was the Root Mean Squared Error after smoothing and normalizing. With this measurement the set of real incidents is first smoothed to create a smooth surface map with crime densities. Both this map and the forecast are then normalized and then RMSE is used to calculate the difference between the normalized maps by comparing them pixel by pixel. This results in a very sensitive performance measure, which is useful for the optimization process. A disadvantage of this abstraction is that the specific locations of the incidents are smoothed, and therefore become less precise. An advantage is that a threshold can be used to ignore locations where very few crimes take place (so called no-man’s-land). When the majority of an area is no-man’s-land this could greatly influence the performance measurement, while the difference between forecasting no crime or very little crime is not very important in practice.

The second selected measurement was the hit rate measure with a threshold. The hit rate measure counts what percentage of all incidents happened in the forecasted hotspot(s). A threshold is used to decide what is a hotspot and what is not. To optimize the threshold for the hit rate the accuracy concentration was used as proposed by Johnson et al. (2009).

Each model is compared to all other models using the cross-validated paired t-test. For these tests 25 folds were used.

3. Optimization

The parameters of the techniques that were selected for comparison had to be optimized to make sure the difference in performance per technique was not caused by parameter settings.

3.1 Random Walk

The random walk technique was not really optimized, but rather the approach used by the police was adopted. It turned out the police used two different models:

- Select data of the last three months prior to when the forecast is made
- Select data from the same month as to forecast for from the last five years

Because the random walk is a simple technique, both models were used in the comparisons.

3.2 ARIMA

The ARIMA technique has many parameters that needed to be optimized: the orders of the model (how far to look back), whether differencing is needed, and the weights that describe the regression patterns in the specific data. To optimize all these parameters the method was used as proposed by Pieterse (2007). Whether to use differencing or not depends on how high the correlation is between time points. The orders are optimized using the (partial) autocorrelation functions. The weights are optimized using a gradient descent algorithm.

Because all these parameters are data specific they have to be optimized for each forecast separately.

3.3 Fuzzy Nearest Neighbor

For the fNN technique the weights for each attribute, and how many incidents to select (the k), had to be optimized. To optimize the weights an evolution strategy[2] was used. The evolution strategy was chosen because it has proved to work very well on optimization problems (e.g. Cruz and Torres, 2007; Beielstein et al., 2003; Schwefel, 1995). The k was optimized separately from the weights using a simple hill climber.

A 5-fold cross validation was used to prevent overfitting of the parameters. For each of the folds different train and test sets were created by choosing different start and end dates for the datasets. These dataset were created to be as diverse as possible.

The optimization was done for different situations. Per situation four factors were varied: the prediction length (the length of time to forecast for), the horizon (the time between the current date and the forecast period), the kind of incidents (burglary, street robbery, or all kinds combined), and amount of available data. The weights were also optimized separately for a police district and a police team.

The optimized weights for the attributes found by the evolution strategy were used by the hill climber to optimize k.

Table 1. Optimized weights for four situations

	Situation 1	Situation 2	Situation 3	Situation 4
k	89%	72%	70%	86%
Recency	0.11	4.74	3.15	0
Year	0.11	0	0.5	3.7
Quarter	2.53	0	1.2	0
Month	0	0.06	0.15	0
Week	1.21	0	0	1.3
Day of the year	0.11			
Day of the month	1.1			
Day of the week	2.31			
Weekend yes/no	1.43			

Some of the results of the optimization are shown in Table 1[3]. The other columns show the optimized weights per situation. Some cells are left empty when the attribute was not informative for the corresponding situation[4].

From the results of the optimization it proved that k should almost always be high. The optimal setting was on average around 80% of the total data. This can also be seen in the second row in Table 1. No interaction could be found with the factors of a situation, but it looked like shorter prediction lengths and larger horizons performed better with a bit lower k.

From the weight optimization only a few clear patterns could be found. Overall in what month, during which moon quarter an incident happened, and the temperature at the time of the incident had lower weights than average. Only the shift of the day in which an incident happened had overall a higher weight than average. However, the shift of the day was only optimized for very few situations; for situations with a prediction length of a day or larger the shift does not add any additional information. The weight for the recency attribute was sometimes very high, but other time surprisingly low, especially for burglary incidents.

Apart from these, no clear patterns could be found of interactions between aspects of a situation and the optimized weights. In some cases weak patterns may have been present in the optimization results.

3.3.1 Abstracting Rules from the Optimization

To be able to create a set of good weights for any possible situation, without optimizing at run time, the idea was to create rules based on the results of the limited set of optimized situations. In this paper a situation is defined as a combination of four factors: the length and horizon of the forecast, the kind of incidents to forecast, and the amount of available data[5]. For each combination of these factors, in combination with the attributes, trend functions were created based on the data produced by the optimization process. These trend functions could then be used to calculate good weights for any situation, even if the model was never optimized for that specific situation.

However, because the optimization did not result in clear patterns, it was very difficult to create useful trend functions. The majority of the trend functions just averaged the weights of all optimization tests.

Therefore it was doubtful whether the trend functions would work as desired. To test this, two fNN models were used in the comparisons; one that used the trend functions, and one in which all weights, and the k, were optimized for each situation specifically. The latter model also optimized the weights and the k together, to take into account possible interaction effects.

4. Comparison Experiments

After the optimization of the three different techniques, these techniques were compared to see the possible differences in performance.

[2] I will not elaborate on how the algorithm works in this paper. For more information see the introductory book by Eiben & Smith (2003).

[3] For all results see Weitjens (2011)

[4] E.g. information on what day of the week an incident happened isn’t relevant when a forecast is created for a whole week.

[5] The amount of available data depends on other factors, such as the location and the type of crime; one type of crime happens more often than another, and at one location more crime happens than at another.

Table 2. Comparison results. Average results are shown of both error measurements and for both a police district and team.

Situation			**Comparison models**					
Kind of incident	**Prediction length**	**Horizon**	**fNNspecific -fNNtrend**	**fNNspecific-RW3m**	**fNNspecific-RW5y**	**fNNtrend-RW3m**	**fNNtrend-RW5y**	**RW3m-RW5y**
All	Month	1 day	+	+	+	+	=	=
Burglary	Month	1 day	=	+	+	+	+	=
Street robbery	Month	1 day	=	+	+	+	+	=
All	Month	90 days	+	+	+	+	=	=
All	Shift	1 day	=	=	=	=	=	=
All	Week	14 days	=	=	=	=	=	=
Burglary	Day	5 days	=	=	+	+	+	=

4.1 Experimental Design

The following models were compared to each other: two different random walk benchmarks as used by the police, the ARIMA technique, the fNN technique with the optimized weights from trend functions, and the fNN technique with optimized weights for the specific situation.

For the comparison between the models the same factors were varied to test different situations as during the optimization: the prediction length, the horizon, and the kind of incident[1]. For these situations prediction maps were created for both a district as well as a team to see how well the models performed with less data, and more spatial detail.

4.2 Results

In Table 2 a summary is given of the comparison results between the different models[2]. For conciseness the average of the two error measures, and the district and team, is shown. The ARIMA model is left out of this table because it performed worse than all the other models in every single test. The names of the other models are abbreviated to fit into the table: RW3m is the random walk model with data of the last three months, RW5y is the random walk model with data of the same month as to forecast for, fNNtrend is the fNN model using the trend functions, and fNNspecific is the fNN model with weights optimized for that specific situation.

The left three columns show the values for the factors of the test situations. In the second row of the other columns it is shown which two models are compared, and in the rows below that the results are shown per situation. A + sign is used when the first mentioned model performed significantly better than the second mentioned, and a = sign when no significant differences were found. Due to averaging no negative significant differences remained. For the t-test a confidence interval of 90% was used.

When looking at the results, on average the fNN models perform better than the random walk models. Between the two random walk models no significant difference in performance is found. The fNNspecific performs better than the fNNtrend in only a few situations, even though more extensive optimization was used.

Not visible from the table is that, although some models outperform others, the differences in performance are quite low. The differences in performance are on average only 2-10%, depending on the situation.

How well the techniques performed differed a lot per situation. For the situations with longer prediction lengths about 80% of the incidents happened in the forecasted hotspots[3], but for shorter predictions length this dropped to as low as 50%. Also with more specific kinds of incidents and smaller areas the performance was slightly lower.

5. Discussion and Conclusions

5.1 Optimization

An evolution strategy was chosen as optimization algorithm for the fNN for its accuracy. This algorithm needs to do many evaluations during the optimization process. For each evaluation a forecast created by the fNN technique was needed. The production of these forecasts took quite some time to complete, partially because of the use of external applications. The need for many evaluations in combination with the time needed per evaluation caused the optimization process to be quite time consuming. Measures were taken to speed up the process. These measures did indeed reduce the calculation time, but it still was a lengthy process (a couple of hours per situation).

The results from the optimization of the attributes for the fNN presented few clear patterns. A decision that may have influenced these results is the use of 5-fold cross validation. Testing a model on only 5 different time points might be too little to capture the patterns in the entire dataset.

[1] The fourth factor, the amount of available data, is not mentioned because in each test all available data was used.

[2] For all results see Weitjens (2011)

[3] For the police district the hotspots occupied 10% of the total area, and for the police team 20% of the total area.

The optimization process produced some unexpected results. The main finding of the optimization was that the weights did not have that much effect on the performance. A property of the fNN technique is that it is inherently robust with rich data; it will perform reasonably well, even with random weights. In a practical sense this is very useful. However, when trying to optimize the weights, this makes it much more difficult.

The value for the k did have more influence on the performance, but this value became very high in most situations. It could be that the k was so high because incidents with a higher distance in the nearest neighbor space already have less impact on the prediction map, so it is not that important to select only the top incidents with the lowest distance. Another cause could be that the algorithm did not find any strong patterns, and therefore just almost all data was selected.

It could also be that crime is to a large extend randomly distributed in time, and therefore inherently very difficult to forecast. This would also make it very difficult for the optimization to find clear patterns.

Besides the fact that only few patterns were found during optimization, another surprising result was that the weight for the recency attribute was sometimes low. A low weight for the recency attribute would suggest that the (near) repeat victimization is not very apparent in the data. This finding is opposite to the findings in other papers (e.g. Ericsson 1995, Bridgeman & Hobbs 1997, Morgan 2000, Shaw & Pease 2000, 2007). The differences in these findings could be caused by the fact that in the experiments described in those papers all situations had prediction lengths of at least a month. For the situations with the prediction length of a month in this paper the weight of the recency attribute was high more often. However, for burglary incidents the recency attribute was especially low, even though a prediction length of a month was used. In the papers by Ericsson (1995) and Morgan (2000) the research was focused on burglary incidents and they found the exact opposite pattern.

5.2 Comparison

The clearest result of the comparison experiments was that the ARIMA model did not perform well at all. A reason for this could be that for the spatial extension the map had to be divided in cells, and for each of those cells a separate estimation had to be made. If the map was divided in 25 cells, on average only 4% of the total data was available for finding patterns per cell, which made it hard to find stable regression patterns. If trying to forecast for less common types of crimes, or for smaller areas, sometimes as few as a couple of dozen records were available per cell. Also the whole area of a cell got the same estimate, and therefore the forecast had very low spatial detail.

Between the four other models no big differences in performance were found. The reason for this could be that crime at different time points does not differ very much. These small differences make it quite easy to get a reasonably good performance, but hard to get a very good performance.

Even though the differences in errors were small in the comparison experiments, significant differences were present. Overall the two fNN models performed consistently better than the two random walk models. Between the random walk models no significant difference was found. Of the two fNN models, the one optimized extensively for each specific situation did perform better than the one using trend functions. The difference in performance was small though, and the extensive optimization took a lot more time.

The performance of all models was lower when trying to forecast for shorter lengths of time. The cause for this is probably that the test sets became much smaller, and therefore patterns could not be found. For more specific types of crime or smaller areas the performance was also lower. In this case both the train and the test set became smaller because less data was available. In other studies it was also found that quite a lot of data is needed to make reasonable forecasts (Gorr & Harries 2003).

5.3 Conclusions

The goal of this research was to explore the geographical crime forecasting field, and compare several existing techniques for geographical forecasting. From the techniques compared the fuzzy nearest neighbor (fNN) algorithm turned out to perform somewhat better than the random walk method, though the differences were small. The ARIMA with a spatial extension did not perform well at all.

The results suggest to use fNN with extensive optimization for situations that are used often (since it is worth the extra investment of the extensive optimization in that case), and create trend functions for all other situations that might be used occasionally.

6. Acknowledgements

We want to thank the police of the regions Amsterdam Amstelland and Midden-West Brabant for their insights and cooperation.

References

Beielstein, T., Ewald, C., Markon, S., & Markon, O. (2003). Optimal Elevator Group Control by Evolution Strategies. *In Proc. 2003 Genetic and Evolutionary Computation Conf. (GECCO’03)* (pp. 1963-1974). Springer.

Block, C. (1984). *Is Crime Seasonal?* Chicago: Illinois Justice Information Authority.

Box, G. (1976). *Time series analysis: forecasting and contol.* Holden-Day.

Bridgeman, C., & Hobbs, L. (1997). *Preventing Repeat Victimization: The Police Officers' Guide.* Retrieved from Home Office: http://rds.homeoffice.gov.uk/rds/pdfs2/ah310.pdf

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory , 13* (1), 21-27.

Cruz, P., & Torres, D. (2007). Evolution strategies in optimization problems. *Proceedings of the Estonian Academy of Sciences, Physics and Mathematics , 56* (4), 299-309.

Eiben, A., & Smith, J. (2003). *Introduction to Evolutionary Computing.* Berlin: Springer.

Ericsson, U. (1995). Straight from the Horse's Mouth. *Forensic Update , 43*, 23-25.

Farrell, G., & Pease, K. (1994). Crime Seasonality, Domestic Disputes and Residential Burglary in Merseyside 1988-90. *British Journal of Criminology , 34* (4), 487-498.

Field, S. (1992). The Effect of Temperature on Crime. *British Journal of Criminology , 32* (3), 340-351.

Gorr, W., & Harries, R. (2003). Introduction to crime forecasting. *International Journal of Forecasting , 19*, 551-555.

Johnson, S., Birks, D., McLaughlin, L., Bowers, K., & Pease, K. (2007). *Prospective Crime Mapping in Operational Context, Final Report.* Retrieved from Home Office: http://www.homeoffice.gov.uk/rds/pdfs07/rdsolr1907.pdf

Johnson, S., Bowers, K., Birks, D., & Pease, K. (2009). Predictive Mapping of Crime by ProMap: Accuracy, Units of Analysis, and the Environmental Backcloth. *Putting Crime in its Place* , 171-198.

Morgan, F. (2000). Repeat burglary in a Perth suburb: Indicator of short-term or long-term risk? In G. Farrell, & K. Pease, *Crime Prevention Studies Volume 12* (Vol. 12, pp. 83-118). Monsey, NY: Criminal Justice Press.

Pieterse, C. (2007). Predicting criminal incident to support police deployment. s.l.

Rotton, J., & Frey, J. (1985). Air Pollution, Weather and Violent Crimes: Concomitant Time-Series Analysis of Archival Data. *Journal of Personality and Social Psychology , 49*, 1207-1220.

Schwefel, H. (1995). *Evolution and Optimum Seeking.* New York: Wiley.

Shaw, M., & Pease, K. (2000). *Research on Repeat Victimisation in Scotland: Final Report.* TSO: Edinburgh.

Weitjens, B. (2011). *Geopredict: Geographical Crime Forecasting for Varying Situations.* Master thesis, Artificial Intelligence, Free University Amsterdam.

Graph-Based N-gram Language Identification on Short Texts

Erik Tromp E.TROMP@STUDENT.TUE.NL
Mykola Pechenizkiy M.PECHENIZKIY@TUE.NL
Department of Computer Science, Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands

Keywords: language identification, classification, n-gram

Abstract

Language identification (LI) is an important task in natural language processing. Several machine learning approaches have been proposed for addressing this problem, but most of them assume relatively long and well written texts. We propose a graph-based N-gram approach for LI called LIGA which targets relatively short and ill-written texts. The results of our experimental study show that LIGA outperforms the state-of-the-art N-gram approach on Twitter messages LI.

1. Introduction

Motivation. The problem of language identification (LI) is interesting on its own. However, in many practical applications LI can be seen as one of the steps of some larger process. Accurate LI can facilitate use of background information about the language and use of more specialized approaches in many natural language processing tasks dealing with a collection or a stream of texts, each of which can be written in a different language. The multilingual sentiment analysis on social media would be a typical motivating example. In (Tromp, 2011), an extensive experimental study shows that the multilingual sentiment classification can be performed more accurately when the process is split into four steps; LI, part-of-speech tagging, subjectivity detection and polarity detection. This becomes possible because at each step after LI, models that utilize language specific knowledge can be applied. Obviously, if the language of some text is identified incorrectly then this error will effect the forthcoming steps of the multilingual sentiment analysis and thus compromise the use of the relatively complicated four step procedure. Therefore, it is highly desirable to minimize the error of LI. Consider another example of machine translation where the source text's language is not known. The translation can not even commence without first identifying this source language.

Numerous supervised machine learning methods have been proposed for LI (Luca et al., 2008; Kranig, 2006). Several experimental studies reported high accuracy results for different collections of relatively long texts with proper grammar. The most widely accepted approach is to use N-grams (Cavnar & Trenkle, 1994). It was shown to be almost 100% accurate for long texts. For example, in (Cavnar & Trenkle, 1994) the experimental results showed 99.8% accuracy on the collection of documents written in fourteen different languages. The results suggested that high accuracies can be achieve for texts having a text length of at least 400 characters. However when LI is done for documents shorter than 300 characters, accuracies start to decrease much faster (though still pertaining the level of 93%) with respect to relatively longer texts having at least 400 characters.

Over recent years, with the popularity of social media, including Twitter and social networks, and consequently social media data analysis like opinion mining, the need for accurate LI (but now on short and grammatically-ill text messages) has become well-motivated again.

Problem formulation and our approach. We consider LI as a supervised learning task, particularly plain single label multi-class classification. Given some historical or training data in which for each text t we know a label l, the language in which this text is written, our goal is to learn a model such that given some previously unseen text we can say as accurately as possible in which language this text is written. We do not consider cases when for a text written partly in one

Appearing in *Proceedings of the 20^{th} Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).

language and partly in some other language someone would like to get both labels as an output. We also do not consider language groups or any other dependencies between the language labels.

We introduce a Graph-based Approach for LI (LIGA) that allows to learn elements of grammar besides using N-gram frequencies.

Main results. The experimental study we performed with the collections of Twitter messages written in six languages shows that LIGA is statistically significantly more accurate than the existing N-gram based approach regardless of the training set size used to learn the models (95-98% for LIGA vs. 87-93% for N-grams) and that LIGA is less prone to overfitting the sources and domain-specific jargon.

Outline. The rest of the paper is organized as follows. In Section 2, we give an overview of the related work on LI discussing the applicability of different approaches to short, grammatically-ill texts. In Section 3 we introduce LIGA, our approach for LI. Section 4 describes the experimental study in which we compare LIGA with the traditional N-gram method. Finally, Section 5 summarizes our findings and suggests direction for future work.

2. Background and Related Work

In this section we discuss related work on LI with the focus on N-gram based approach as the current state-of-the-art for LI, which we use in LIGA.

2.1. N-Gram-Based Approach

The N-gram-based approach for LI (Cavnar & Trenkle, 1994) chops texts up in equally-sized character strings, N-grams, of length n. It is assumed that every language uses certain N-grams more frequently than other languages, thus providing a clue on the language the text is in. This idea works due to Zipf's law stating that the size of the r-th largest occurrence of the event is inversely proportional to its rank r (Ha et al., 2003). Experimental studies in (Cavnar & Trenkle, 1994) suggest that using trigrams (at the character level) generally yields the best results.

In (Cavnar & Trenkle, 1994) the out-of-placement measure is used to compare unlabeled text against the model. This measure sorts the N-grams in both the model as well as the unlabeled text separately based on their occurrence counts and compares the model's occurrence list with the text's list. Later, in (Ahmed et al., 2004) it was shown that the use of a cumulative frequency based measurement yields similar accuracy results yet is more time efficient. The out-of-placement measure works well when sufficient training data is available whereas the cumulative frequency measurement works equally well with little data at hand. Therefore we will use the cumulative frequency measurement in our experiments.

2.2. Other Approaches

The idea behind the N-gram-based approach is borrowed from (Dunning, 1994) where using Markov Models for LI was considered. This approach however lacks the intuition the N-gram approach has and requires more time for training a model and for classifying a new text.

A similar straightforward approach is to use word frequencies. One variant is to use short words (Prager, 1999) as they occur regularly in a language and usually differ per language. Another variant is to use the most frequently occurring words (Martino & Paulsen, 1996; Cowie et al., 1999) for the same rationale.

The compression-based approach for LI was proposed in (Harper & Teahan, 2001). Labeled data is compressed using so-called prediction by partial matching (PPM) approach to construct language models. An unlabeled text is also compressed and the number of bits required to encode this new document is compared to the number bits used in the language models. The likeliness of a text with a language model is computed using entropy as a similarity measurement.

3. LIGA - Graph-Based N-gram Language Identification

In our approach we want to utilize not only word presence and occurrences but also their ordering. To capture the ordering of words, we create a graph model on labeled data. The labels of its vertices represent the presence of words in a given language. The weights of the vertices represent the frequencies of words in a given language. The crucial part is in the presence and weights of the edges, which try to capture the grammar (in this particular case only the word ordering) of a language.

As a starting point for our method, we use the N-gram-based approach described in Section 2.1. We will thus not truly capture word information but N-gram information. Next, we give the preliminaries (Section 3.1), the methodology to learn LIGA and to use it for the model (Section 3.2) and to classify unlabeled texts (Section 3.3), and time and space complexity analysis (Section 3.4).

3.1. Preliminaries

We extend a basic graph $G = (V, E)$ with a labeling function $\mathcal{L} : V \to L$. This labeling function assigns to each vertex $v \in V$ a label $l \in L$ uniquely identifying the vertex. Let *Lang* denote all languages present in our training set, then the function $\mathcal{W}_v : V \times Lang \to \mathbb{N}$ assigns for each vertex $v \in V$ and every language $l \in Lang$ a weight. For edges we have a similar function $\mathcal{W}_e : E \times Lang \to \mathbb{N}$. Since we will incrementally construct our graph, we may encounter that, for a label l of a vertex u to be added, $\exists_{v \in V} : v \neq u \wedge \mathcal{L}(v) = l$. We then say that $\mathcal{L}(v)$ is *defined*. We say $\mathcal{L}(v)$ is *undefined* in all other cases. We use the same notion of *defined* for $\mathcal{W}_v$ and $\mathcal{W}_e$.

Using the mentioned extensions, we represent a graph as the following quintuple.

$$G = (V, E, \mathcal{L}, \mathcal{W}_v, \mathcal{W}_e)$$

A labeled text t of which the language l is known is denoted as the binary tuple (t, l). An unlabeled text s of will be denoted as the binary tuple (s, λ). We denote all N-grams of a text t as the ordered list $N_{n_t} = [g_1, g_2, ..., g_k]$ where n denotes the length of the N-grams. The order in N_{n_t} respects the order in which the N-grams occur in (t, l).

3.2. Learning a Model

Our goal is given a training set $\mathcal{T}$ consisting of labeled texts for every language in *Lang* to learn a model consisting of a single graph G.

For each text $(t, l) \in \mathcal{T}$ we construct the list N_{n_t}. For every $m \in N_{n_t}$ we create a vertex v with $\mathcal{L}(v) = m$, but only if $\mathcal{L}(v)$ is *undefined*. For every $e \in \{(u, v) \in V \times V : (\mathcal{L}(u) = m_i \wedge \mathcal{L}(v) = m_{i+1}) \Rightarrow (m_i, m_{i+1} \in N_{n_t})\}$ we create an edge e, but only if $e \notin E$. The weights are updated as follows:

$$\mathcal{W}_v(v, l) = \begin{cases} \mathcal{W}_v(v, l) + 1 & \text{if } \mathcal{L}(v) \text{ is } \textit{defined} \\ 1 & \text{otherwise} \end{cases}$$

$$\mathcal{W}_e(e, l) = \begin{cases} \mathcal{W}_e(e, l) + 1 & \text{if } \mathcal{W}_e(e, l) \text{ is } \textit{defined} \\ 1 & \text{otherwise} \end{cases}$$

When we add a node or edge (i.e. when $\mathcal{L}(v)$ or $\mathcal{W}_e(e, l)$ is *undefined* respectively), we initialize the weights of all languages for that vertex or node to 0 before applying the weight updates. When applying the aforementioned definitions for all $(t, l) \in \mathcal{T}$ we get our graph $G = (V, E, \mathcal{L}, \mathcal{W}_v, \mathcal{W}_e)$. We illustrate this by an example. Consider the following two texts of which the first is in Dutch (NL) and the second is in English (EN).

$$\begin{aligned} (t_1, NL) &= \quad \textit{is dit een test} \\ (t_2, EN) &= \quad \textit{is this a test} \end{aligned}$$

We then first create the ordered lists of N-grams. When using trigrams ($n = 3$) we get the following, where a space is denoted by a dot $\cdot$.

$N_{3_{t_1}}$ = [is·, s·d, ·di, dit, it·, t·e, ·ee, een, en·, n·t, ·te, tes, est]

$N_{3_{t_2}}$ = [is·, s·t, ·th, thi, his, is·, s·a, ·a·, a·t, ·te, tes, est]

We next start constructing the graph. For each $n \in N_{3_{t_1}} \cap N_{3_{t_2}}$ we add a vertex v to our graph, having $\mathcal{L}(v) = n$. For example, for the first element in $N_{3_{t_1}}$ we will create the vertex v having $\mathcal{L}(v) = is\cdot$. For the first element in $N_{3_{t_2}}$ we will not add a new vertex as $\mathcal{L}(v)$ is defined. In our example, for vertex v (having $\mathcal{L}(v) = is\cdot$) we will have $\mathcal{W}_v(v, NL) = 1$ and $\mathcal{W}_v(v, EN) = 1$ as $is\cdot$ occurs once in both the Dutch as well as the English text.

We next add edges. We will have edges from for example v to u ($e = (v, u)$) where $\mathcal{L}(v) = is\cdot$ and $\mathcal{L}(u) = s \cdot d$, capturing the order between the first and second elements of $N_{3_{t_1}}$. Since this connection occurs only once, and only for the Dutch text, we have that $\mathcal{W}_e(e, NL) = 1$ and $\mathcal{W}_e(e, EN) = 0$.

Figure 1 shows the graph resulting from this example. The labels of the vertices are shown at the topmost position inside the vertices. The weights per language are listed with the vertices and edges.

3.3. Classifying a Text

Once we have constructed G, we can use it to classify unlabeled texts. To do so, we first need to transform an unlabeled text into something similar to G such that we can compare the two.

While for constructing G we use weights to indicate multiple occurrences of a given N-gram, for the unlabeled text we create multiple vertices – one for every occurrence. We thus in fact get a simple graph, a path $\pi = (V, E, \mathcal{L}, v_{start})$. Here $|V| = |N_{n_t}|$ and if $N_{n_t} = [n_1, n_2, .., n_k]$ then $E = \{(u, v) | \mathcal{L}(u) = n_i \wedge \mathcal{L}(v) = n_{i+1}\}$. The node $v_{start} \in V$ is the starting node of our path. To illustrate this, we consider the following (Dutch) text *is dit ook een test* of which we would not know the label in advance. The path π for this example is shown in Figure 2.

A crucial step is to compare π and G. We compute the so-called *path-matching scores* for each language $l \in Lang$. Conceptually, this is done by 'laying' the path over the graph and measuring its similarity for each language. Since all languages we have knowledge

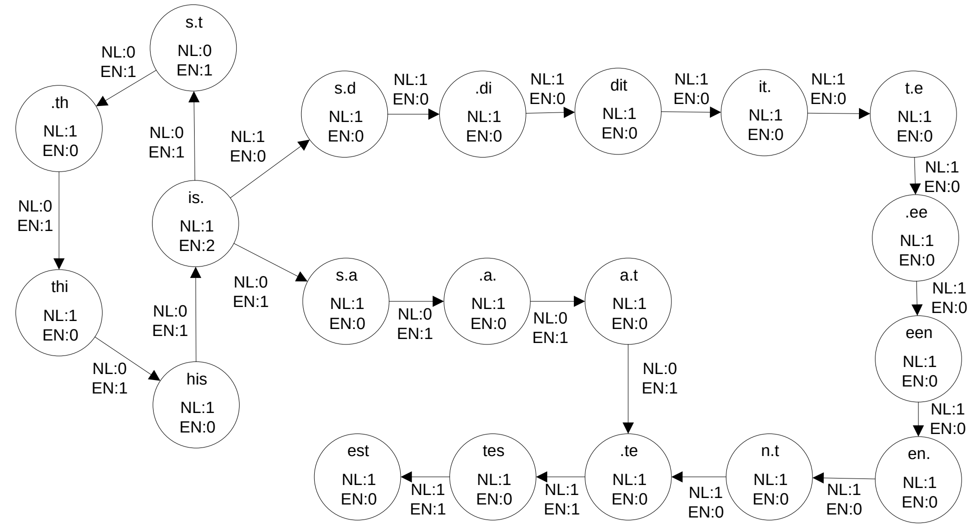


Figure 1. The graph resulting from the example

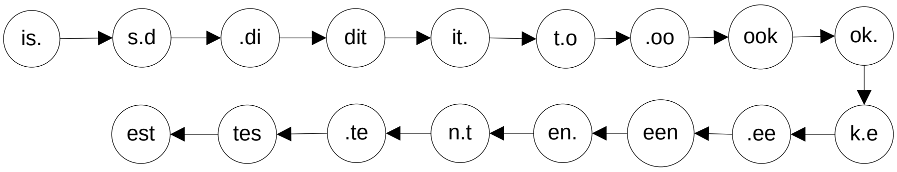


Figure 2. The path resulting from the unlabeled example

of are present in a single model, we can compute the scores for all languages in one go.

To keep track of the matching scores of our languages, we maintain a scoring function $\mathcal{PM} : Lang \rightarrow \mathbb{R}$ assigning a rational number to every language $l \in Lang$. Initially, we say that $\mathcal{PM}(l) = 0$ for all $l \in Lang$. Let $G = (V, E, \mathcal{L}, \mathcal{W}_v, \mathcal{W}_e)$ be the graph representing our model and $\pi = (V', E', \mathcal{L}', v_{start})$ be our labeled path. In order to keep our method robust with respect to differing quantities of labeled data for different languages, we will normalize the scores. Let $\Sigma_v = \Sigma_{v \in V}(\Sigma_{l \in Lang}(\mathcal{W}_v(v, l)))$ be the total sum of all weights contained in all nodes in G. Also, let $\Sigma_e = \Sigma_{e \in E}(\Sigma_{l \in Lang}(\mathcal{W}_e(e, l)))$ be the sum of all weights contained in all edges in G. We then start traversing our path π in G starting with node v_{start}. Let v'_i denote the node we are currently visiting (initially, $v'_i = v_{start}$). We try to find a $v \in V$ such that $\mathcal{L}(v) = \mathcal{L}'(v'_i)$. Note that there is at most one such node but there may be none. We update the matching scores according to (1). In this step we account for the occurrence of a given N-gram from π in G.

The next step is to account for the order. We find the only edge $e' \in E'$ that has our previous node v'_i as source, if any (since the last node has no outgoing edges). We thus find the edge $e' = (v'_i, v'_{i+1})$. We then update the matching scores according to (2).

We have now accounted for the order between two N-grams present in our path whose label is given in nodes v'_i and v'_{i+1}. We next continue traversing our path by performing (1) again for v'_{i+1} after which we can apply (2). This process continues until we find a node $v'_{end} \in V'$ such that $\neg(\exists_{v' \in V'} : (v'_{end}, v') \in E')$; this is the ending node not having any outgoing edges.

When we have matched the entire path onto the graph, we need our function $\mathcal{PM}$ to determine the language. For each language, $\mathcal{PM}$ maps it onto a value in $\mathbb{R}$. More specifically, since the weights of the nodes accounted to $\mathcal{PM}$ are normalized and so are the weights of the edges, we end up with a score in $[0, 2]$ (when our model is isomorphic to our path for a given language, we get a score of 1 for the nodes and 1 for the edges). We now say that the text of which we were identifying the language is in language $l = argmax_{l \in Lang} : \mathcal{PM}(l)$.

$$\forall_{l \in Lang} : \mathcal{PM}(l) = \begin{cases} \mathcal{PM}(l) + \frac{\mathcal{W}_v(v,l)}{\Sigma_v} & \text{if } \exists_{v \in V} : \mathcal{L}(v) = \mathcal{L}'(v'_i) \\ \mathcal{PM}(l) & \text{otherwise} \end{cases} \quad (1)$$

$$\forall_{l \in Lang} : \mathcal{PM}(l) = \begin{cases} \mathcal{PM}(l) + \frac{\mathcal{W}_e(e,l)}{\Sigma_e} & \text{if } \exists_{e \in E} : (\exists v, w \in V : \mathcal{L}(v) = \mathcal{L}'(v'_i) \wedge \\ & \quad \mathcal{L}(w) = \mathcal{L}'(v'_{i+1}) \wedge e = (v, w)) \\ \mathcal{PM}(l) & \text{otherwise} \end{cases} \quad (2)$$

3.4. Time and Space Complexity of LIGA

Let $\mathcal{T} = \{(t_1, l_1), (t_2, l_2), .., (t_m, l_m)\}$ represent all texts t_i with their respective labels l_i of our training set. For every t_i we create its N-grams which can be done in $\mathcal{O}(|t_i|)$ time, i.e. linear in the size of the text. As we create the N-grams, we add them to our graph in parallel by keeping track of the previous N-gram for the edge addition. The time required to build G is hence $\mathcal{O}(|t_i|)$ for a single text t_i. Let now $t_{max} = argmax_{t_i \in T} : (\forall_{t_j \in \mathcal{T}} : |t_i| \geq |t_j|)$ be the longest all texts in $\mathcal{T}$. The total time required to train a model is bound by $\mathcal{O}(m \cdot |t_{max}|)$, where m is the number of texts in $\mathcal{T}$, which is asymptotically equivalent to the regular N-gram-based approach. Note however that the constant for LIGA is greater than for the N-gram-based approach.

When we classify (t_{new}, λ) with G, we again first create the N-grams in $\mathcal{O}(|t_{new}|)$ time, forming a path. We then traverse the path in our graph. Since for every N-gram in our path we have to check at most one node in G (namely, the node having the same label) and at most one edge, this is also linear in the size of the unlabeled text. Classifying a text hence requires $\mathcal{O}(|t|)$ time.

For the space required to store our model, the worst case would be when we have a completely non-overlapping training set. Every text in such a training set would yield a single string or path that is disconnected from all other paths. Since such a single path represents a single text t, we have one node for each N-gram of the text, thus requiring $\mathcal{O}(|t|)$ space. We will have exactly $|t| - 1$ edges since the starting N-gram has no incoming edges and the ending N-gram has no outgoing edges. The space required to store the edges is hence also linear in the text's size, requiring $\mathcal{O}(|t|)$. Using similar notation as with the time complexity, we need $\mathcal{O}(|l| \cdot m \cdot |t_{max}|)$ space to store the model. The $|l|$ component is the number of languages present in our model and originates from the fact that we need to store $|l|$ weights for each node and edge. Similar reasoning shows that we require $\mathcal{O}(|t|)$ space to store the path representing unlabeled data.

4. Experimental evaluation

We compare the performance of the proposed LIGA approach with the N-gram-based approach for LI. Particularly, we are interested in the following three aspects; (a) comparing the accuracy of two approaches on the LI of *short texts* (Twitter messages), (b) looking into the effect of reducing the amount of training data on the accuracies, and (c) comparing the generalization and (over)specialization properties of two approaches with respect to the source, domain or jargon the present in the training data.

In their work (Cavnar & Trenkle, 1994) use newsgroup articles having culture as topic. The messages in their dataset hence all share the same domain. The generalization/specialization experiments address this problem.

In the following sections we first describe the dataset used in this study and the experiment setup, and then present the main results.

4.1. Dataset and Experiment Setup

The dataset was constructed from the social medium *Twitter*. The Twitter API is used to extract messages from accounts known to only contain messages of a specific language. We do this for six languages and six accounts per language. The six languages are German, English, Spanish, French, Italian and Dutch. These are languages we have sufficient knowledge of to identify. Moreover, including Spanish, French and Italian presents a challenge as these languages contain a lot of similar word extensions and trigram patterns. For every language we have at least one account of a person instead of an institution (such as BBC News). We assume that each account has its own domain and hence its own jargon that typically is not, or less often, used in the other domains. When we indicate a domain as *random* we mean that its messages are a mixture of other domains. The rationale behind using six accounts per language is that our experiments require us to have different domains but incorporating tens or hundreds of accounts is very laborious.

As a pre-processing step, we inspect our data as in (Cavnar & Trenkle, 1994), removing messages that

Table 1. Splitting the data into training and test sets.

Lang.	Domain	Exp. 1		Exp. 2
		s_1	$train_1$	$test_{2_{gen}}$
DE	Sports	...		
		s_n	$test_1$	
		n_1	$train_1$	$train_2$
	National news	...		
		n_n	$test_1$	$test_{2_{spec}}$
		r_1	$train_1$	$test_{2_{gen}}$
	Random	...		
		r_n	$test_1$	
NL	Telecom.			
...	...	...		...

Table 2. Accuracies averaged over 50 runs.

Experiment	LIGA	N-gram	t-Test
5% sample	94.9 ± 0.8	87.5 ± 1.5	✓
10% sample	96.4 ± 0.5	90.6 ± 1.0	✓
25% sample	97.3 ± 0.5	92.5 ± 0.9	✓
50% sample	97.5 ± 0.5	93.1 ± 0.8	✓
5% vs. 50%	94.9 ± 0.8	93.1 ± 0.8	✓
Generalization	92.4 ± 1.0	83.5 ± 1.8	✓
Specialization	98.3 ± 0.8	95.5 ± 1.6	✓
One holdout	95.6 ± 1.6	89.2 ± 4.3	✓
Two holdouts	95.2 ± 0.9	88.1 ± 2.7	✓

contain multiple languages or bilingual terminology. From each message we also deliberately remove links, usernames preceded by an @ sign, term references preceded by a # sign or smilies such as *:)* and punctuation. The rationale behind this is that we want to learn a model on the language itself whereas these entities are language-independent. Moreover, the use of Twitter for our experiments is just to show an application on short texts, learning Twitter-specific patterns containing username or channel references is not desired. The final dataset contains 9066 labeled messages of at most 140 bytes.[1]

In both approaches we use trigrams (as suggested in (Cavnar & Trenkle, 1994)) and the frequency based measurement of (Ahmed et al., 2004).

In all cases we compare the mean accuracy of the N-gram approach against the mean accuracy of LIGA. The mean accuracies are obtained by taking the mean of 50 different ten-fold cross-validation experiment runs. We check for statistical significance using pairwise T-tests.

In order to compare the accuracies of both methods, we learn a model on one part of the data, which we call $train_1$, and test it on another part of the data, called $test_1$ formed as illustrated in Table 1, column Exp. 1. We use all accounts of all languages. The size of $train_1$ varies to investigate the influence of the corpus' size. We use 5%, 10%, 25% and 50% of the entire dataset stratified per language and sampled uniformly.

We also investigate how much each approach learns about the actual language itself rather than about a particular domain. To analyze this, we learn a model on data from one domain and test it on other domains. For each language, we choose a single account on which we learn our model and then test on the remaining accounts (Table 1, column Exp. 2). The training set $train_2$ consists of $\frac{2}{3}$ of all data of one single account for each language. There are now two test sets. The first, $test_{2_{spec}}$ is the remaining $\frac{1}{3}$ of the same account and will allow us to judge how specific each model is for a single domain. The second, $test_{2_{gen}}$ consists of all other accounts. This test set will show us how well the learnt model generalizes to other domains.

Finally, we analyze the influence of jargon. To study this, we learn a model on all accounts of all languages except for one or two hold-out accounts which are reserved for testing. Assuming that each account represents a single domain, we can get to know how important it is to capture domain-specific jargon since any domain-specific jargon present in one of the holdout accounts likely will not be included in our model. The formation of the training and test datasets can be seen as the inverse of Table 1, column Exp. 2.

4.2. Experiment Results

The main results are presented in Table 2 which shows, averaged over 50 experiment runs, the accuracies and standard deviations for LIGA and N-gram approaches. The last column shows the result of pairwised T-test (which was positive for each comparison in our case).

Effect of the training set size. As expected we can see that as the size of the training data grows, the accuracy increases and variance decreases with both approaches. However, LIGA has much better performance than N-gram approach especially when a small amount of labeled data is available.

The row '5% vs. 50%' of Table 2 compares LIGA

[1] The dataset is made available online at `http://www.win.tue.nl/~mpechen/projects/smm/`

learned on 5% of the training data against the N-gram-based approach using 50% of the training data. LIGA still has statistically significant higher accuracy.

It can be clearly seen from Figure 3 that LIGA *consistently* outperforms N-gram approach on each of the 50 experiment runs. To avoid overloading of the graph we present only accuracies corresponding to the use of 5% and 50% of available labeled data.

Using more than 50% of labeled data for training a LI model in our case did not result in any further significant improvement of the classification accuracy for either of the two approaches and therefore we omit these results for the sake of conciseness.

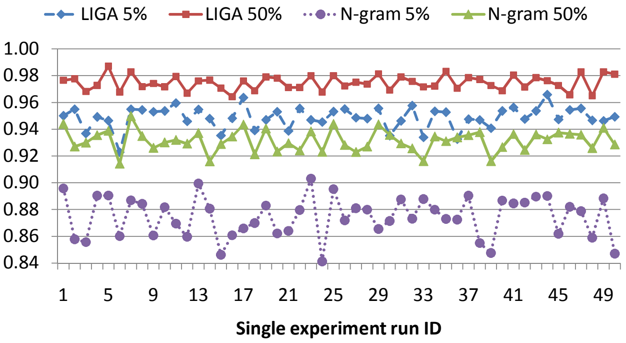


Figure 3. Accuracy results over 50 different runs for LIGA and N-gram approaches when trained either on 5% or on 50% of available labeled data.

Generalization and domain-specific jargon results. The 'Generalization' and 'Specialization' rows in Table 2 and Figure 4 show the results from the generalization experiments. As one would expect, a model learned on a specific domain yields higher accuracies in that domain than in other domains. This behavior is shown by the difference in accuracies between the generalization results and the specialization results. The specialization accuracy for both approaches is often close to 100%, yet LIGA is statistically significantly better than N-gram approach.

For the generalization experiments, we clearly see that LIGA again consistently (Figure 4) outperforms the N-gram based approach. The accuracy of LIGA never drops below 90% whereas that of the N-gram based approach is never higher than 90%.

LIGA achieves higher accuracies both within the domain the model was trained on as well as outside that domain. This indicates that LIGA is likely to be less sensitive to domain-bound usage of language and hence is likely to learn more about the language itself rather than about the domain messages belong to.

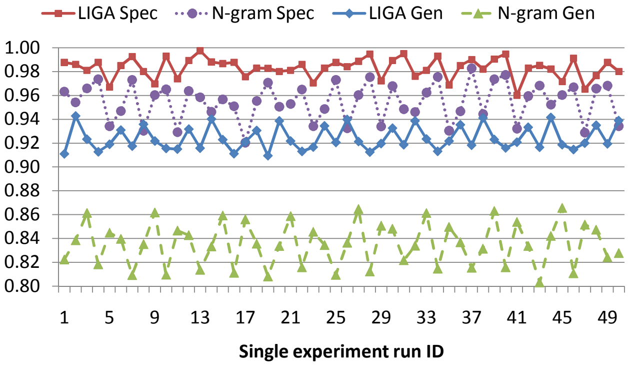


Figure 4. Accuracy results over 50 different runs for LIGA and N-gram approaches when trained on one domain and tested on all other domains (see Table 1, column Exp. 2 for clarification).

The results plotted in Figure 5 stem from holding out one or two accounts. Averaged results are in the last two rows of Table 2. We can observe a moderate drop in accuracy for both approaches with respect to not holding out any account. LIGA's accuracy drops by 2-3% on a single holdout and the N-gram-based approach's accuracy drops by 4-5%. Testing on two holdout accounts decreases accuracy a bit further but not much. This indicates that the use of domain-specific jargon introduces an error of around 3% for LIGA and 5% for the N-gram-based approach. LIGA consistently outperforms the N-gram-based approach.

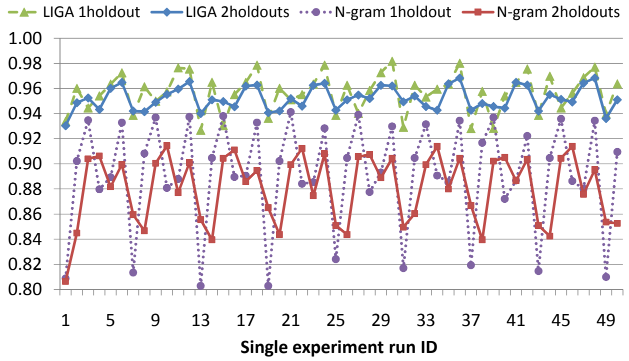


Figure 5. Accuracy results over 50 different runs for LIGA and N-gram approaches with one and two hold out accounts.

We observe that for some holdout accounts the accuracy is always lower than for others. There are thus some domains that use more jargon than others. However, across multiple experiment runs, the variance in accuracy is always much larger for the N-gram approach.

5. Conclusions

In this paper we have studied the problem of language identification on relatively short texts typical for social media like Twitter. Earlier works on language identification showed promising and highly accurate results for well-constructed, sufficiently long enough texts. However, the results were shown to deteriorate considerably when texts become a few hundred characters long.

We proposed LIGA – a graph-based approach that aims to capture the elements of language grammar. The experimental results confirmed our expectation that for short texts LIGA outperforms the current state-of-the-art N-gram approach for language identification, showing consistently higher accuracy results even with an order less labeled data used to learn the model from. Besides that, our experiments suggested that LIGA is less likely to be sensitive to use of jargon or to domain boundaries.

Future Work. LIGA currently captures only one aspect of grammar, the order of words or trigrams at the character level. It would be interesting to see what other grammatical aspects can aid to the problem of language identification. One such aspect may be to incorporate words that can (or often do) start sentences. Our method can be easily extended to use this information by computing N-grams that begin some texts and taking them into account in path-matching.

We evaluated the performance of our approach only on short texts extracted from Twitter with respect to the goals of this study. However, it would be interesting to compare these results with results obtained from using longer texts or texts extracted from other sources. In addition to regarding other sources, using more data and especially incorporating more languages, gives stronger results and a broader comparison.

Language identification is typically part or one step of a bigger process. The error made in this step is hence propagated along the pipeline of remaining tasks. It is Interesting to study how severely this error affects succeeding steps. In (Tromp, 2011) we show how to quantify this effect for the case of the multilingual sentiment analysis on social media.

An open issue not addressed in related work on language identification is dealing with an absence of certain N-grams or words that influences the certainty of classification. When an unlabeled text contains a number of N-grams not present in a learned model, the learned model will not be confident about the label. This suggests assigning confidence scores to the assigned labels that can be utilized in the further natural language processing routine or in the mechanism suggesting updates to or relearning of the language identification models.

References

Ahmed, B., Cha, S., & Tappert, C. (2004). Language identification from text using n-gram based cumulative frequency addition. *Proc. CSIS'04.*

Cavnar, W., & Trenkle, J. (1994). N-gram-based text categorization. *Proc. 3rd Symp. on Document Analysis and Information Retrieval (SDAIR-94).*

Cowie, J., Ludovic, Y., & Zacharski, R. (1999). Language recognition for mono- and multilingual documents. *Proc. of the Vextal Conference.*

Dunning, T. (1994). Statistical identification of language. *TR-MCCS-94-273, New Mexico State Univ.*.

Ha, L., Sicilia-garcia, E., Ming, J., & Smith, F. (2003). Extension of zipfs law to word and character n-grams for english and chinese. *Journal of Computational Linguistics and Chinese Language Processing.*

Harper, D., & Teahan, W. (2001). Using compression-based language models for text categorization. *Proc. Workshop on Language Modeling and Information Retrieval.*

Kranig, S. (2006). Evaluation of language identification methods. *Proc. ACM SAC 2006.*

Luca, E. D., Grothe, L., & Nuernberger, A. (2008). A comparative study on language identification methods. *Proc. 6th Conf. on International Language Resources and Evaluation.*

Martino, M., & Paulsen, R. (1996). Natural language determination using partial words. *US Pat. 6216102 B1.*

Prager, J. (1999). Linguini: Language identification for multilingual documents. *Proc. 32nd Hawaii Int. Conf. on System Sciences.*

Tromp, E. (2011). Multilingual sentiment analysis on social media. Master's thesis, Dept. Computer Science, Eindhoven University of Technology.

Visualizations of Machine Learning Behavior with Dimensionality Reduction Techniques

Keywords: visualization, machine learning, dimensionality reduction

Bo Gao BO.GAO@CS.KULEUVEN.BE
K.U.Leuven - Department of Computer Science, Celestijnenlaan 200A, 3001 Leuven, Belgium

Joaquin Vanschoren JOAQUIN@LIACS.NL
Leiden University - LIACS, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands

Abstract

There are many different approaches to machine learning, and each approach has its own characteristics and behavior. In order to investigate the aspects of these approaches, large amounts of machine learning experiments with high dimensionality(data characteristics, algorithm characteristics, parameters settings, evaluation metrics, etc.) are generated and collected within databases, such as the Experiment Database. To enable the user to gain insight into this mass of meta-data about machine learning algorithms efficiently and effectively, different Dimensionality Reduction techniques are investigated. Based on this, a visualization tool is built to help users analyze the behavior of learning algorithms. The experiment results of these techniques on different meta-datasets are discussed in this paper.

1. Introduction

The Experiment Database (ExpDB) is a public database designed to collect and organize large numbers of past experiments donated by many researchers (Vanschoren et al., 2011). Each experiment may contain different algorithms, different parameter settings and different datasets, etc. We call the data that describe the details of the experiments meta-data. Unfortunately, the high dimensionality of the meta-dataset becomes a prohibiting factor for us to gain insights in the algorithms. Indeed, many trends may occur in a (curved) low-dimensional subspace (or manifold) of the data. In other words, the data points may lie close to a manifold of much lower dimensionality than that of the original data space (Bishop, 2006). As such, we look into different Dimensionality Reduction (DR) techniques which aim to map the data from a high dimensional space to a low dimensional one (typically 2D or 3D), so that an overview picture of the data is presented to the researcher. As the saying goes: a picture is worth a thousand words. Large amounts of data can be imprinted on a single image that is far more comprehensible. Furthermore, new knowledge can be perceived and discovered through visualizations. Therefore the researcher should be able to use the visualizations to analyse machine learning behavior efficiently.

Hence, we can summarize our research goal as twofold: First, we aim to investigate the relationships between datasets and machine learning algorithms. For instance, which parameter setting of which algorithm can achieve high performance (e.g. predictive accuracy) on what kind of datasets, or which type of learning algorithms (e.g. rule-based or decision trees or kernel methods) are more robust on certain datasets. We also hope that through visualizations, new relationships can be discovered. Second, we want to evaluate different state-of-the-art DR techniques in the context of machine learning meta-datasets.

In the remainder of this paper, we discuss the key implemented DR techniques in Section 2, and show the results of the evaluations and visualizations of the DR techniques on machine learning meta-dataset(s) in Section 3. Finally, in Section 4, we conclude.

Appearing in *Proceedings of the 20^{th} Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by Authors/Owners.

2. Dimensionality Reduction

In this section, we discuss a few representative DR techniques, including the traditional linear as well as several state-of-the-art non-linear ones. Afterwards, we give a brief overview of all implemented DR techniques. Because of the scope of this paper, we try to keep the mathematical equations to a minimum throughout the text, and emphasize on the concepts and intuitions behind these techniques.

A note on notations: given a dataset, we use N to denote the number of instances (rows), with i being the index for each instance ($i = 1, ..., N$). In the original high dimensional space, D denotes dimensionality, or the number of attributes (columns), and X_i denotes the i_{th} instance (row) of the dataset. In the low dimensional space, M denotes the number of attributes after mapping ($M < D$), and Y_i denotes the i_{th} instance of the mapped dataset.

2.1. Linear DR Techniques

2.1.1. PCA

Principal Component Analysis (PCA) can be formulated as the orthogonal projection of the original data onto the (linear) *principal subspace* such that the original variance in the data is maximally preserved. The algorithm performs eigenvalue decomposition on the $D \times D$ covariance matrix of the original dataset (Bishop, 2006). The obtained M eigenvectors with the largest eigenvalues are called the *principal components*(PC's) of the original dataset. The low dimensional data is the projection of the original data onto the PC's. PCA has several limitations. First, it assumes the linearity of data, whereas in real world, the intrinsic structure of the data can be so "curvy" that a "rigid" subspace (comprised of a set of orthogonal PC's) becomes an inadequate approximation. Second, it assumes a single Gaussian distribution of the data. Third, it focuses on preserving the largest variances, which makes the approximation vulnerable towards noise. Many variants of PCA have been proposed, such as Principal Curves (Hastie & Stuetzle, 1989), Probabilistic PCA (Tipping & Bishop, 1999), Kernel PCA (Schölkopf et al., 1998), etc. From these variants, only Kernel PCA is currently implemented in the visualization tool.

2.1.2. CLASSICAL SCALING

Classical Scaling (Cox & Cox, 1994) (CS) is a type of MDS (MultiDimensional Scaling). The aim of MDS is to preserve the pairwise distances as much as possible in the mapping. The low dimensional data can be derived via eigenvalue decomposition on the distance matrix. It is shown that CS is intrinsically the same as PCA (Van der Maaten et al., 2009). Like PCA, CS also assumes the data to be linear. It also ignores the neighborhood information of a data point and large distances are preserved better than small ones causing the detailed information among "close" data points is overlooked. Other (non-linear) types of MDS are proposed to address these weaknesses of CS, such as Isomap and Sammon Maps, which will be discussed in Section 2.2.3 and 2.2.6 respectively.

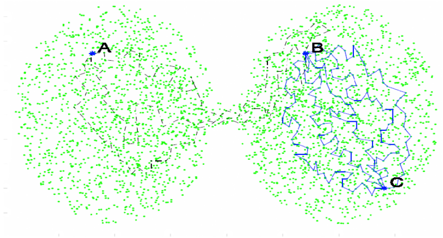


Figure 1. An illustration of Markov random walks in Diffusion Maps

2.2. Non-Linear DR Techniques

In order to find the intrinsic, often non-linear manifold embedded in the original data space, besides constructing a mixture of linear models (such as Probabilistic PCA), we have an alternative: to consider a single non-linear model. Many techniques have been invented or adapted in this way, yielding distinctive merits. We implemented the following representative state-of-the-art non-linear DR techniques in the visualization tool.

2.2.1. DIFFUSION MAPS

Diffusion Maps build a probabilistic model with pairwise *transition probabilities* to reflect the degree of closeness between each pair of data points (e.g. points A and B). The higher this probability is, the easier it becomes to "walk" from A to B (Lafon & Lee, 2006). As shown in Figure 1, the given set of data points (green dots) form two clusters, data point A is in the left cluster, B and C are in the right cluster. Since there are more paths between B and C than between B and A, it will be easier for B to "walk" to C than to A. Based on the transition probabilities, *diffusion distances* (Lafon & Lee, 2006) are defined between pairs of data points. Diffusion Maps aim to find a low dimensional representation in which the Euclidean distances between pairs of data points match the diffusion distances as well as possible. This is achieved by eigenvalue decomposition on the transition probability matrix $\mathbf{P^{(t)}}$ (Lafon & Lee, 2006).

2.2.2. KERNEL PCA

Kernel PCA(KPCA) is a non-linear extension of PCA, which transforms the original data into a higher-dimensional feature space using a kernel function (Schölkopf et al., 1998) and performs regular PCA in the feature space to obtain "curved" principal components. The low dimensional representation can be found through the eigenvalue decomposition on the kernel matrix of the original data. The main limitation of KPCA is the selection of an appropriate kernel and the parameter configuration of that kernel. Different methods have been proposed to solve the kernel-selection problem, e.g. hold-out testing (Golub et al., 1979), semi-definite programming (Graepel, 2002), etc. Still, these methods are computationally expensive (Van der Maaten et al., 2009).

2.2.3. ISOMAP

Isometric Feature Mapping (Isomap) (Balasubramanian & Schwartz, 2002) is similar to Classical Scaling, but instead of a Euclidean distance matrix, Isomap uses a geodesic distance matrix. The geodesic distance between two data points is the accumulative distance of the shortest path between the two points found by Dijkstra's Algorithm (Dijkstra, 1959) on a graph. However, the geodesic distance is vulnerable towards "short-circuiting": where the data points that are far away from each other are taken as neighbors.

The neighborhood graph is constructed as follows. First, a Euclidean distance matrix is constructed as in Classical Scaling. Second, a neighborhood graph is constructed based on the Euclidean distances, in which only data points considered to be neighbors are connected, and each connection is assigned a weight. There are two approaches to find a data point A's neighbors: (1) If the Euclidean distance between A and a point B is smaller than a predefined threshold ε, then B is A's neighbor, and a connection is assigned to A and B (this is called ε-Isomap); (2) Rank the Euclidean distances between A and all the other data points, select the K nearest points as A's neighbors ($K < m-1$, this is called k-Isomap). We will call this k-Isomap in further discussion.

2.2.4. LLE

Locally Linear Embedding (LLE) (Roweis & Saul, 2000) also constructs a (k-nearest neighbor) graph representation of the manifold of the original data. But in contrast to Isomap, which uses geodesic distances to characterize the global geometry of the manifold, LLE focuses on preserving the local geometry. We can imagine that a data point and its k neighbors form a local plane: the data point would become the plane's topological center, and the plane is unique to the data point. A set of reconstruction weights is defined so that each data point X_i can be represented as a linear combination of its k neighbors (Eq.1).

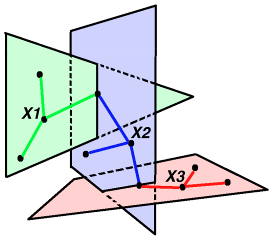


Figure 2. Example sub-planes in LLE, the data points X_1, X_2 and X_3 are the topological centers for each plane

$$X_i \approx w_{i,1}X_i^{(1)} + w_{i,2}X_i^{(2)} + ... + w_{i,k}X_i^{(k)} \quad (1)$$

with

$$\sum_{j=1}^{K} w_{i,j} = 1. \quad (2)$$

We call $w_{i,j}$ the reconstruction weights. An illustration is shown in Figure 2, in which three sub-planes in the original data are identified by the three data points X_1, X_2, X_3 and their respective neighbors. The reconstruction weights are then derived by solving a linear system (Saul & Roweis, 2000) to minimize the cost function:

$$\phi_1(W) = \sum_{i=1}^{N} \left| X_i - \sum_{j=1}^{k} w_{i,j}X_i^{(j)} \right|^2 \quad (3)$$

Because the reconstruction weights are invariant to translation, rotation and rescaling, we have:

$$\phi_2(Y) = \sum_{i=1}^{N} \left| Y_i - \sum_{j=1}^{k} w_{i,j}Y_i^{(j)} \right|^2 \quad (4)$$

The low dimensional representation is derived via the eigenvalue decomposition on the sparse matrix (Roweis & Saul, 2000): $(I-W)^T(I-W)$ to minimize the cost function (Eq.4), I is an $N \times N$ identity matrix.

2.2.5. LAPLACIAN EIGENMAPS

Laplacian Eigenmaps are similar to LLE in the sense that they both try to preserve the local geometrical properties of the manifold of data and they both use the sparse weight matrix based on a neighborhood graph. After the neighborhood graph is constructed, which is the same as that in Isomap and LLE, a weight matrix is directly computed based on a kernel function (E.g. a Gaussian kernel in Eq.5):

$$w_{i,j} = \left\{ \begin{array}{cc} e^{-\frac{|X_i - X_j|^2}{2\sigma^2}} & if\, neighbor(X_i, X_j) \\ 0 & else \end{array} \right. \quad (5)$$

A cost function (an sum of the weighted distances between a data point and its k nearest neighbors) is then minimized:

$$\phi(Y) = \sum_{i=1}^{N}\sum_{j=1}^{N} \left(|Y_i - Y_j|^2 w_{i,j} \right) \quad (6)$$

The Gaussian kernel function emphasizes the small distances between data points more than the large distances. In other words, the closer neighbors contribute more to the cost function than the farther ones. The cost function can be minimized through a eigenvalue decomposition on the matrix $D^{-1}L$, with D being the diagonal matrix $D_{i,i} = \sum_{j=1}^{N} w_{i,j}$, W being the sparse weight matrix and $L = D - W$ (Belkin & Niyogi, 2001). The neighborhood graph-based approaches have several limitations: the construction of the neighborhood graph is susceptible to overfitting, and the local linearity assumption is susceptible to discontinuities in the manifold of data.

2.2.6. SAMMON MAPS

Sammon Maps is another type of MDS. In contrast to the DR techniques discussed previously, Sammon Maps (Sammon, 1969) do not perform eigenvalue decomposition on a (transformed) proximity matrix to minimize a cost function and find the low dimensional coordinates of the original data. Instead, it tries to find the low dimensional mapping through an iterative process.

$$\phi(Y) = \frac{1}{\sum_{i<j} d(X_i, X_j)} \sum_{i<j} \frac{(d(X_i, X_j) - d(Y_i, Y_j))^2}{d(X_i, X_j)} \tag{7}$$

In each iteration, the error (Eq.8) is computed:

$$e_{i,j} = \gamma \frac{(d(X_i, X_j) - d(Y_i, Y_j))}{d(Y_i, Y_j)} (Y_i - Y_j) \tag{8}$$

The low dimensional coordinates are updated iteratively:

$$Y_i^{new} = Y_i + e_{i,j} \tag{9}$$

$$Y_j^{new} = Y_j - e_{i,j} \tag{10}$$

with $\gamma > 0, i, j \in [1, N]$, γ is the learning rate.

Being a non-spectral technique, Sammon Maps are less susceptible to high dimensionality. The Sammon cost function is shown in Eq.7, which is similar to that of Classical Scaling. However, one possible limitation of Sammon Maps is weight $1/d(X_i, X_j)$ can lead to overfitting.

2.3. An Overview of the DR techniques

As illustrated in Figure 3: from the eight DR techniques we have discussed, PCA and Classical Scaling are linear, the rest are non-linear. Among the non-linear ones, Kernel PCA, Diffusion Maps and Laplacian Eigenmaps (LE) utilize kernel functions. LE, LLE and Isomap are all based on neighborhood graphs, in which LE and LLE use a sparse proximity matrix, while Isomap uses a full one. We can also distinguish Sammon Maps from the others by its non-spectrality.

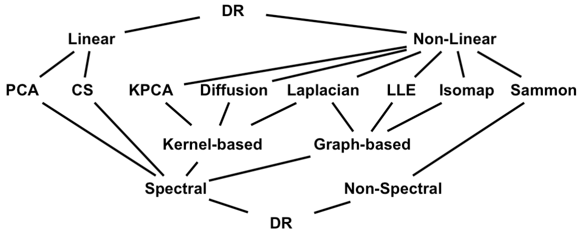


Figure 3. Categorization of DR techniques.

Table 1. The attribute description of ***bagging***.

Name	Description
1.nr_iterations	number of bagging iterations
2.baseLearner	base learner used
3.dataset	name of the dataset
4.classCount	number of classes
5.nr_attributes	number of the attributes
6.nr_sym_attributes	number of symbolic attributes
7.nr_num_attributes	number of numeric attributes
8.lm_naiveBayes	performance of naive bayes landmarker
9.lm_1nn	performance of 1-nearest neighbor landmarker
10.lm_1rule	performance of 1-rule landmarker
11.nr_examples	number of data points
12.nr_missingValues	number of missing values
13.class_entropy	entropy of the class attribute
14.default_accuracy	default accuracy of the dataset
15.evaluation	predictive accuracy of bagging

3. Experiments

In this section, we apply the eight DR techniques with different parameter settings to two machine learning meta-datasets: ***bagging*** and ***algo-performance***, and discuss the results.

3.1. Datasets

bagging contains 1437 runs of the standard Bagging algorithm, with different parameter settings and base-learners, on different classification UCI[1] datasets. The attributes of the dataset are summarized in Table 1. Note that landmarkers are simplified algorithms used to characterize a dataset (Pfahringer et al., 2000). Attributes 2 and 3 are categorical, and are used as color labels in the visualizations, the rest are numerical, which serve as the input to DR techniques.

algo-performance contains the predictive accuracies of 293 algorithm-parameter combinations on 83 different UCI datasets (D=83). E.g. combination SVM-C-1.0-Polynomial-E-3 is a support vector machine with complexity constant 1.0 and a polynomial kernel with power 3. Each row is a different algorithm-parameter combination (all the algorithms are from Weka[2]) and each column is the performance on a specific dataset.

[1] http://archive.ics.uci.edu/ml/

[2] http://www.cs.waikato.ac.nz/ml/weka/

3.2. Evaluations

3.2.1. TRUSTWORTHINESS AND CONTINUITY

When we reduce the dimensionality of the data, the topological properties of the data in the original space will not be completely preserved, which leads to distortions. There are two types of distortions: (1) Data points that are originally far away from each other are mapped close to each other in the low dimensional space. (2) Data points that are originally close to each other are mapped far away instead. In order to measure the quality of a dimensionality reduction, we use Trustworthiness and Continuity (Venna & Kaski, 2006) to characterize the first and the second distortion respectively. Trustworthiness is defined as follows:

$$Trustworthiness\,(k) = 1 - A\,(k) \sum_{i=1}^{N} \sum_{j \in U_k} (r_{i,j} - k) \quad (11)$$

with

$$A\,(k) = \begin{cases} \frac{2}{N \cdot K \cdot (2N - 3k - 1)} & if\ k < \frac{N}{2} \\ \frac{2}{N \cdot (N-k) \cdot (N-k-1)} & if\ k \geq \frac{N}{2} \end{cases} \quad (12)$$

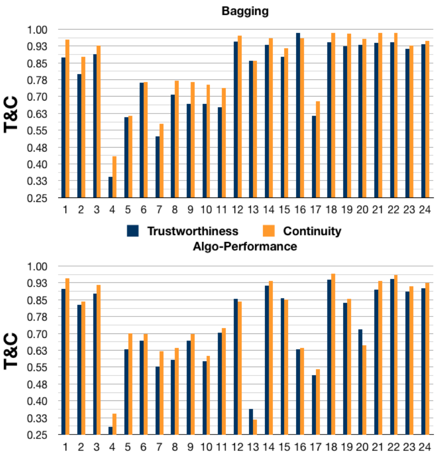
In which N is the number of data points, k is the predefined number of neighboring points, and $r_{i,j}$ is the rank of the data point j according to the data point i in the original space: the closer j is to i, the lower $r_{i,j}$ will be. The ranks are natural numbers. U_k is the set of data points that are within the k-nearest neighbors of data point i in the low dimensional space but do not appear in the original space. $A(k)$ is a scaling factor that scales the second term, i.e. the error term. The more data points with high ranks (the data points that are far away from each other) are wrongly mapped close to each other in the low dimensional space, the larger the error term will be. Thus, the trustworthiness ranges from 0 to 1, 1 means completely trustworthy, 0 means completely untrustworthy. Similarly, Continuity is defined as following:

$$\text{Continuity}\,(k) = 1 - A\,(k) \sum_{i=1}^{N} \sum_{j \in V_k} (\hat{r}_{i,j} - k) \quad (13)$$

$\hat{r}_{i,j}$ is the rank of data point j according to data point i in the low dimensional space: the closer j is to i, the lower $\hat{r}_{i,j}$ will be. V_k is the set of data points that are within the k-nearest neighbors of data point i in the original space but do not appear in the low dimensional space. When data points are neighbors in the original space but not in the visualization, this will increase the error. Continuity also ranges from 0 to 1, a larger number means a better continuity.

3.2.2. RESULTS

For each dataset, the Trustworthiness(T) and Continuity(C) with 6 different k-values[3] are measured for each DR technique. The T and C scores are then computed for each dataset. The resulting graphs are shown in Figure 4: the table on the right denotes the indices of the DR techniques with their particular parameter settings. We use Gaussian kernels in Diffusion Maps, Kernel PCA and Laplacian Eigenmaps, and also use Sigmoid kernels in Kernel PCA for comparison. As we can see in Fig.4, for ***bagging***, DR No.12 has relatively high Continuity, though not the best, and it has high Trustworthiness comparing with other DR with high Continuity. Also note that DR No.16, 21 and 22 are also very good visualization candidates on ***bagging***. For ***algo-performance***, DR No.22 achieves the highest Trustworthiness and Continuity. As such, we will discuss the visualizations of DR12: Diffusion Maps ($t = 1, \sigma = 1$) and DR22: PCA, i.e. 22 on the two datasets respectively. The visualizations of KPCA(gauss, $\sigma = 1$) on ***bagging*** and Sammon Maps($i = 50$) on ***algo-performance*** are also shown in section 3.3.3 to give the reader an impression of other DR techniques. Due to the space limitations, we will not discuss the latter two visualizations in detail.



1	Classical Scaling
2	Isomap(k=12)
3	Isomap(k=36)
4	Isomap(k=6)
5	LLE(k=12)
6	LLE(k=36)
7	LLE(k=6)
8	Laplacian(k=12, σ=1)
9	Laplacian(k=12, σ=0.2)
10	Laplacian(k=12, σ=5)
11	Laplacian(k=36, σ=1)
12	Diffusion(t=1, σ=1)
13	Diffusion(t=1, σ=0.2)
14	Diffusion(t=1, σ=5)
15	Diffusion(t=6, σ=1)
16	KPCA(gauss, σ=1)
17	KPCA(gauss, σ=0.3)
18	KPCA(gauss, σ=11)
19	KPCA(sigmoid, γ=1, c=0)
20	KPCA(sigmoid, γ=1, c=10)
21	KPCA(sigmoid, γ=0.3, c=0)
22	PCA
23	Sammon(i=10)
24	Sammon(i=50)

Figure 4. The experiment results of different DR techniques on the machine learning meta-datasets.

3.3. Visualizations

3.3.1. ON THE *BAGGING* DATASET

First, we apply the diffusion maps technique on the *bagging* dataset. The result is shown in Figure 5. In each sub-figure, a different dimension is used to color the data points. In Figure 5.2, we color the three base learners IBk, J48 and Random Tree with red, green and blue color respectively. For categorical values, each category gets a unique color, and for numerical values, points are colored ranging from red (high) to blue (low).

It is interesting to see which attributes correlate with

[3] k is selected so that three are smaller than $N/2$ and the other three are larger than $N/2$, N being the number of instances in the dataset.

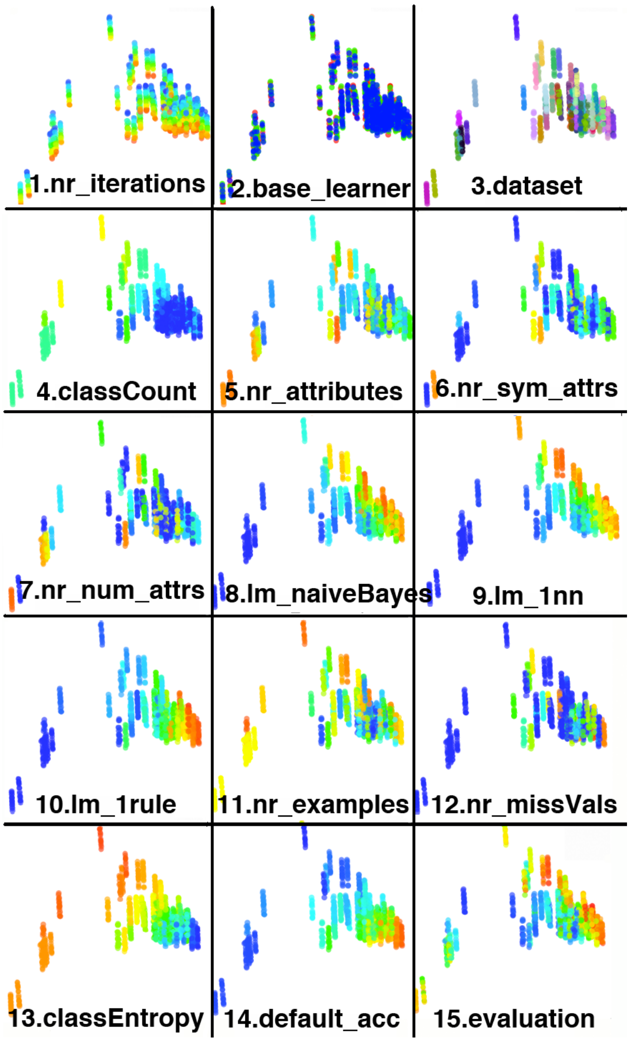


Figure 5. Diffusion Maps ($\sigma = 1, t = 1$) on the *bagging* dataset, colored based on the 15 attributes. For categorical values, same category has the same color which is distinguishable from other categories. For numerical values, color range "red..orange..yellow..green..blue" indicate a series of values from being high to being low.

the dimensions generated by diffusion mapping. The Y-axis correlates with the number of bagging iterations (Figure 5.1), while the X-axis correlates with the landmarking results (Figure 5.8-10). The latter roughly indicate how easily the data can be modeled with simple classifiers. Since the data can be most accurately mapped to these two dimensions, they have a large impact on the distribution of the data.

When looking at the actual clusters that formed, it is clear that the data is clustered by dataset. This is not surprising, since the data contains a lot of attributes for dataset characteristics. Figure 6 shows these clusters in more detail, with labels for several datasets. Still, we also see more general clusters, which seem to correlate with the class count (Figure 5.4) and class entropy (Figure 5.13). For instance, there is one dense cluster on the right for datasets with few classes.

Another interesting discovery is that the landmarkers seem to correlate well with the final evaluation of the experiment, but only on the generally 'easier' datasets on the right. On the leftmost datasets, landmarkers perform badly (the blue dots in Figure 5.8-10), while the evaluation of the complete learning algorithms is generally good. Especially the fact that these are datasets with a low default accuracy, high class entropy and higher class count leads us to believe that the "simple" models generated by landmarkers are just too simple to capture the structure in the data, even if complete versions of the algorithm can do this. Also note that the different landmarkers disagree on the top-most datasets (Figure 5.8-10), which tend to have many classes. Clearly, some landmarkers are more robust to many classes than others.

When looking at Figure 5.2, we see that diffusion mapping does not separate the different base learners. This is not surprising as there are no numerical attributes that describe the base-learning algorithm used. Measurable numerical properties of the learning algorithms should be added to the data so that they may be included in the diffusion mapping.

The evaluations on the same datasets tend to have similar scores: we can see them forming small clusters in the plot as shown in Figure 6. Still, some clusters (datasets) show a vertical gradient, indicating the effect of the number of iterations. The effect is not as pronounced as we expected, but this is probably due to the fact the results for different base-learners overlap with each other. Also, on some datasets the Bagging algorithms can achieve particularly high or low scores, e.g. the dataset Sick always has high evaluations with the Bagging algorithms, whereas the dataset Abalone always has low evaluations.

Finally, all this shows that dimensionality reduction techniques are extremely useful to study the performance of algorithms under many different variables: different parameter settings, different datasets, and

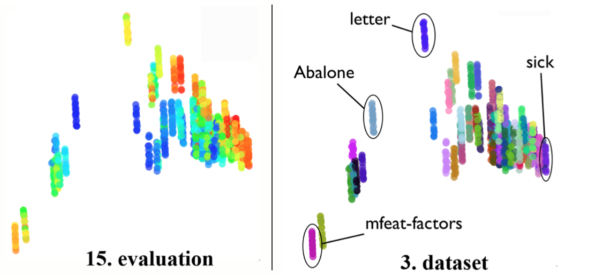


Figure 6. Diffusion Maps ($\sigma = 1, t = 1$) on the *bagging* dataset, a detailed illustration.

different properties of the algorithms and datasets. The high number of attributes can successfully be reduced to a representation which retains most of the information and immediately allows us to look for interesting patterns on sight.

3.3.2. ON THE *ALGO-PERFORMANCE* DATASET

Second, we apply the PCA technique on the *algo-performance* dataset.

The result is shown in Figure 7. In Figure 7.1, we can find some outliers such as "'Raced-Incremental-Logit-Boost-default" and "Bagging-I-1..25-ZeroR", which are far away from the "big cluster". This indicates that these algorithm-parameter combinations perform very differently from their peers on the given datasets. Also, when combining with Figure 7.2 and 7.3, we discover that the outliers perform poorly on the *letter* and *anneal* datasets, since the dots representing them are colored blue. On the other hand, the combinations in the"big cluster" generally have a high performance on the two datasets. One step further, when coloring the algorithm-paramenter combinations based on their general categorization (the combinations belonging to SMO (Support Vector Machines) are blue, the ones belonging to Bagging are green and the others are red), we can see that SMO methods mostly occupy the top of the visualization, and looking back to Figure 7.2 and 7.3, we discover that SMO methods perform very well on the *letter* dataset but very poorly on the *anneal* dataset.

Finally, let's look at the small clusters formed in the outliers. With the help of the interactive function in the visualization tool, we see that the Bagging algorithms with the base learners HyperPipe and ConjunctiveRule form their own small clusters in the outliers, whose performance is poor on the given datasets. We also discover that the Bagging algorithms with the base learners Multi-layer Perceptron (MLP) and J48 have much better performance. Many more patterns can be discovered in the visualization.

3.3.3. OTHER DR VISUALIZATIONS ON *BAGGING* AND *ALGO-PERFORMANCE*

From Figure 8 we see a similar distribution of points as Figure 5. The points are vertically aligned according to the number of iterations of bagging algorithms, and horizontally aligned according to the general 'difficulty' of the dataset.

From Figure 9 we see a similar distribution of points as Figure 7. The major cluster as well as small peripheral clusters can be identified.

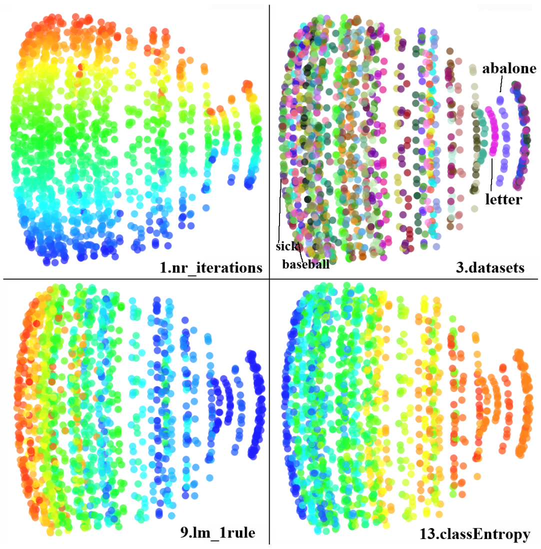


Figure 8. KPCA (gauss, $\sigma = 1$) on the *bagging* dataset, colored based on attributes 1, 3, 9 and 13.

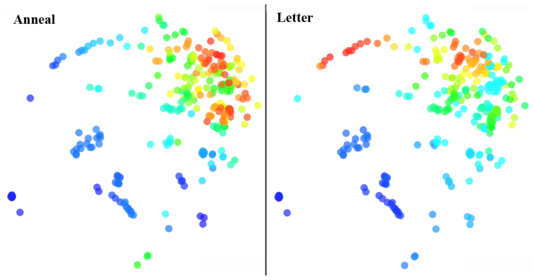


Figure 9. Sammon Maps ($i = 1$) on the *algo-performance* dataset, colored based on attributes 'anneal' and 'letter'.

4. Conclusions

In this paper, we investigated different dimensionality reduction (DR) techniques for visualizing high dimensional meta-data on machine learning algorithms, retrieved from the Experiment Database (ExpDB). First, we provided an overview of interesting DR techniques. Next, we evaluated these DR techniques on trustworthiness and continuity to select the adequate techniques for visualizing two different sets of meta-data.

We observed that the same DR techniques with different parameter settings can lead to significantly different results, such as Isomap with k = 6 and k = 36. Kernel and/or probabilistic techniques (such as Kernel PCA and Diffusion Maps) achieve more satisfying results than graph-based techniques (such as LLE and Isomap) on the two meta-datasets. Especially sparse graph-based techniques, such as LLE, do not perform well. Indeed, in order to achieve high performance with kernel-based techniques, the type of the kernel and the corresponding parameters have to

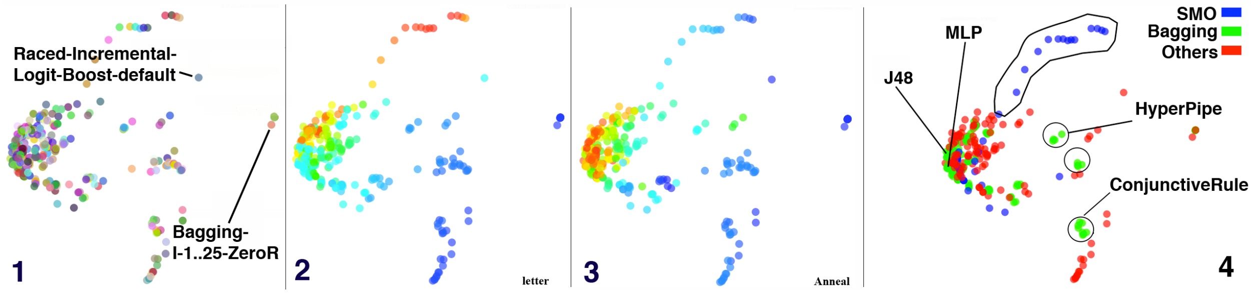


Figure 7. PCA on the *bagging* dataset. The first three sub-figures are colored based on (1) the categorical attribute "dataset", the performance of all the algorithm-parameter combinations on (2) the *letter* dataset and (3) the *anneal* dataset. The forth sub-figure is colored based on a general categorization of the algorithm-parameter combinations.

be adjusted several times manually, which is rather inconvenient. Also, we find that linear DR techniques have stable (and sometimes even better) performance compared to their non-linear peers. Non-spectral techniques (namely Sammon Maps) generally give good results. Furthermore, similar to kernel-based techniques, the other nonlinear DR techniques require parameter optimization.

Determining which technique is preferable depends not only on parameter configuration, but also the given datasets. Selecting or synthesizing appropriate techniques based on a given dataset is an interesting direction to look into in future work. For example, one could build a model that selects an appropriate DR technique with appropriate parameter(s) for a given dataset based on Trustworthiness and Continuity. Furthermore, other DR techniques, especially deep-structured ones are worth investigation. Next, we could use DR techniques to compare how algorithm performance on the UCI 'benchmark datasets' compare to performance results on real-world datasets. We can also store the results of the experiments of DR on machine learning meta-data (let's call them DR meta-data to distinguish with ML meta-data) into a database (e.g. ExpDB) so that we can investigate their performance in detail.

Finally, the produced DR visualization tools is of course also applicable for any other high dimensional data.

References

Balasubramanian, M., & Schwartz, E. L. (2002). The Isomap Algorithm and Topological Stability. *Science*, *295*, 7.

Belkin, M., & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems 14* (pp. 585–591). MIT Press.

Bishop, C. M. (2006). *Pattern recognition and machine learning (information science and statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Cox, T., & Cox, M. (1994). *Multidimensional scaling*. London: Chapman & Hall.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*, 269–271. 10.1007/BF01386390.

Golub, G., Heath, H., & Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, *21*, 215–224.

Hastie, T., & Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, *84*, 502–516.

Lafon, S., & Lee, A. (2006). Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *28*, 1393 –1403.

Pfahringer, B., Bensusan, H., & Giraud-Carrier, C. (2000). Meta-learning by landmarking various learning algorithms. *In Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 743–750). Morgan Kaufmann.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, *18*, 401–409.

Saul, L. K., & Roweis, S. T. (2000). *An introduction to locally linear embedding* (Technical Report).

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, *10*, 1299–1319.

Tipping, M. E., & Bishop, C. M. (1999). Mixtures of probabilistic principal component analysers. *Neural Computation*, *11*, 443–482.

Van der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: A comparative review. *TiCC-TR 2009-005*.

Vanschoren, J., Blockeel, H., Pfahringer, B., & Holmes, G. (2011). Experiment databases: A new way to share, organize and learn from experiments. *Machine Learning*.

Venna, J., & Kaski, S. (2006). Local multidimensional scaling. *Neural Networks*, *19*, 889–899.

Full Papers

Visual Data Mining for Higher-level Patterns: Discrimination-Aware Data Mining and Beyond

Keywords: visualization, discrimination, privacy, awareness, data mining

Bo Gao BO.GAO@CS.KULEUVEN.BE
K.U.Leuven - Department of Computer Science, Celestijnenlaan 200A, 3001 Heverlee, Belgium

Bettina Berendt BETTINA.BERENDT@CS.KULEUVEN.BE
K.U.Leuven - Department of Computer Science, Celestijnenlaan 200A, 3001 Heverlee, Belgium

Abstract

An important question facing visualization methods is how to be both general and support open-ended exploratory analysis. In this paper, we propose a visualization approach that can on the one hand be applied to any (classification or association) rules, but that is suited to bringing out features of mined patterns that are especially important in discrimination-aware and privacy-aware data mining. We define new interestingness measures for items and rules and show various ways in which these can help in highlighting information in interactive settings. We conclude by arguing how this approach can lead to a new generation of feedback and awareness tools.

1. Introduction

The essence of data mining is to make "hidden" relationships in data explicit and to find actionable patterns. One example are classification rules that suggest a course of action based on a set of items (attribute-value combinations). To be able to understand these patterns better, derive new knowledge or new hypotheses, it is often helpful to have a meta-level view: which rules are similar to which other ones (and how), which items are highly predictive of a certain outcome (and in interaction with which other items), etc. Visualization is a prime technique for achieving such condensed, meta-level representations of mining results and can therefore be invaluable for applications in business, medicine, and other areas.

The need to inspect mining results carefully for such meta-level relationships between features and outcomes becomes even stronger when specific data, rules and other patterns become the object of scrutiny: The flipside of data mining is that it may make relationships visible that various stakeholders do not wish to become explicit, and that the patterns it finds may suggest actions that various stakeholders do not wish to be taken. Such concerns may lead to a new approach to keep and/or treat these data as private.

Closely linked to questions of *privacy* is the occurrence of *discrimination*: the treatment taken toward or against a person of a certain group in consideration based solely on the category of this group, such as gender, age or nationality. We will investigate *discrimination-aware data mining* as a technological solution to identify discriminatory classification rules. We will argue why this approach has a direct application for exploring knowledge and uses of other types of data. Specifically, in a field whose semantics are as ill-defined and shifting as those of privacy, tools that couple data-mining, visualization and interactivity are very well-suited for giving feedback to users and raising awareness of the consequences of their actions.

The main contribution of this paper is a set of measures (Section 3) and visualizations (Section 4) that allow the analyst to focus on important patterns and their properties. These include similarities on rules and visualizations of items based on interestingness scores, applicable to the analysis of *any* rule set. We illustrate the advantages of our approach by concentrating on the *specific* example of the visualization of the results of discrimination-aware data mining (dis-

Appearing in *Proceedings of the 20^{th} Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the authors.

cussed along with other related work in Section 2). Concluding, we elaborate on generalizations, in particular for feedback and awareness tools (Section 5).

2. Related Work

2.1. Discrimination-aware Data Mining

Pedreschi, Ruggieri and Turini (Pedreschi et al., 2008; Ruggieri et al., 2010a) introduced the problem of discrimination-aware data mining. Here, discrimination is the legally non-permitted use of data on specific demographics as the basis for a decision. They study instances of *directly discriminatory classification rules* (e.g., not giving a credit because of the applicant is a foreigner.) and of *indirectly discriminatory classification rules*, which at first sight look innocuous but can be shown to imply discrimination via an inference (such as not giving a credit because the applicant lives in a certain ZIP code, when it is known that most people living there are migrants).[1] Their main point is that a mining analysis (a) may reveal that in past data, regularities may be found that appear to signal discrimination: for example, that in a group of people all characterized by itemset X, non-single women were "bad credits" more often than the group as a whole. However, (b) this deviation should be considered "grave" only if the additional information makes the class assignment much more likely than based on X alone. Here, "much more" is operationalized via a set of interestingness measures (building on the *lift* of rules) and thresholds, which declare rules to be discriminatory only if they exceed this threshold. This formalizes the legal notion of discrimination as a "disproportionate burden" put on certain demographic groups. For example, the U.S. Equal Pay Act states that "a selection rate for any race, sex, or ethnic group which is less than four-fifths of the rate for the group with the highest rate will generally be regarded as evidence of adverse impact". Discrimination-aware data mining has has since been extended in various directions, cf. (Pedreschi et al., 2009; Calders & Verwer, 2010; Hajian et al., 2011).

2.2. DCUBE: a textual interface for discrimination-aware data mining

The DCUBE system[2] (Ruggieri et al., 2010b) implements the (Ruggieri et al., 2010a) approach using the Apriori algorithm for extracting rules and an Oracle database for storing them. Like DCUBE, we use the *German Credit* public domain dataset (Frank & Asuncion, 2010) for demonstration. Its attributes comprise various demographics (gender, marital status, nationality, ...) and details of the applicant's existing property and loan purposes. The Credit Class indicates whether a loan was given or not.

In DCUBE, the user can declare items as Potentially Discriminatory (PD) and set other parameters[3] before the rule extraction process. For example, many laws forbid discrimination based on nationality, gender, marital status or age, which in the German Credit dataset are expressed by items such as $foreign\_worker = yes$, $personal\_status = female\_div\_or\_sep\_or\_mar$[4] and $age = gt\_52d6$[5]. The remaining items are automatically taken as PND (Potentially Non-Discriminatory). The classification rules are then mined through both Direct and Indirect Discrimination analyses (DD resp. IDD), as shown in Figure. 1. DD analysis yields rules in which an outcome (such as $credit = bad$) follows from potentially discriminatory items (usually in conjunction with PND items), while in IDD analysis, it follows from innocuous-looking PND items, which are shown to lead to inferences towards DD rules via background knowledge.

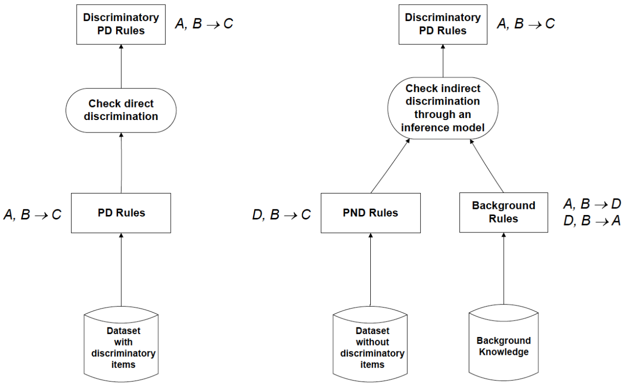

Discriminatory PD Rules
A, B → C
Check direct discrimination
A, B → C
PD Rules
Dataset with discriminatory items
Discriminatory PD Rules
A, B → C
Check indirect discrimination through an inference model
D, B → C
PND Rules
Background Rules
A, B → D
D, B → A
Dataset without discriminatory items
Background Knowledge

Figure 1. Modeling the process of direct (left) and indirect (right) discrimination analysis (Pedreschi et al., 2008).

Using SQL queries, the user can extract a number of PD rules with defined constraints. A PD rule takes the form: $A, B \rightarrow C$, with A as PD item(s), B some PND item(s), and C the class item. Rules resulting from DD or IDD analysis are ranked with respect to the interestingness measures that show the "degree of

[1] Inferences leading to the use of personal data for non-intended or non-permitted uses are a problem also addressed in privacy research; see (Berendt et al., 2008) for a formalization and a tool for avoiding such uses in an online-analytics system.

[2] http://kdd.di.unipi.it/dcube

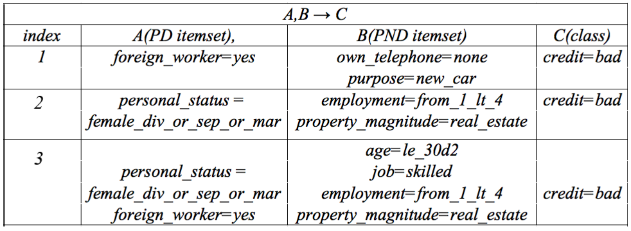
[3] max/min support, max size of frequent itemsets, ...

[4] a currently divorced, separated or married woman

[5] a person older than 52.6 years.

discrimination", which will be discussed in Section 3. Table 1 shows a list of PD rules.

Table 1. Example List of PD Rules.



A,B → C			
index	*A(PD itemset),*	*B(PND itemset)*	*C(class)*
1	*foreign_worker=yes*	*own_telephone=none* *purpose=new_car*	*credit=bad*
2	*personal_status = female_div_or_sep_or_mar*	*employment=from_1_lt_4* *property_magnitude=real_estate*	*credit=bad*
3	*personal_status = female_div_or_sep_or_mar* *foreign_worker=yes*	*age=le_30d2* *job=skilled* *employment=from_1_lt_4* *property_magnitude=real_estate*	*credit=bad*

There are two limitations regarding the usability of DCUBE. First, the analysis via SQL queries is fully functional but appropriate for professional/technical users and much less accessible for other users. Second, the large number of rules makes it difficult to obtain an overview and interpret the patterns. For example, it would be difficult to find out how frequent items are within rules, which two (or more) items are closely related to one another, which cluster of rules has significant discriminatory measures, etc. To gain a more comprensive perspective on the whole set of rules, we need to employ Information Visualization.

2.3. Visualizations of Association Rules

Many visualization solutions have been proposed for association rules. For example: the 2D matrix with 3D cubes (Wong et al., 1999), in which the x and y-axes of the 2D matrix denote premises and conclusions of the rules respectively, the heights of the 3D cubes denote the interestingness measures of the rules, e.g. confidence. Alternatively, we can use each column of the 2D matrix as a rule and each row as an item. If a rule contains a certain item, this item's space in the matrix will be occupied with a 3D cube, whether this items belongs to the rule's premise or conclusion will be differentiated with color coding on the cube. Another example are the directed graphs of (Blanchard et al., 2007a), in which rules are represented as graphs. Each node is an item or a combination of items (Blanchard et al., 2007b), arrows are used to link from premises to conclusions. Other examples are the Mosaic Display (Blanchard et al., 2007a) and ARVis (Kuntz et al., 2000; Blanchard et al., 2007b), etc. These visualizations can clearly express each rule's internal structure, but are insufficient to provide higher-level knowledge, especially when the number of rules gets large. The distribution of the rules and items or the inter-relations among them cannot be seen easily. Our approach to visualize classification rules (as a special case of association rules) targets these shortcomings.

3. Interestingness and Relatedness Measures for Items and Rules

The goal of our visualization is to highlight particularly "discriminatory" items on the one hand and to show relationships between particularly relevant (i.e. discriminatory) rules on the other hand. We therefore define new measures of interestingness based on those measuring the degree of discrimination in discrimination-aware data mining.

We start from the discriminatory measure (*D-measure*) on individual rules, an output of DCUBE. The larger the *D-measure* of a rule is, the more certain we are of this rule being discriminatory against people in disadvantaged groups, such as foreign workers, single mothers, etc. Selection lift (*slift*) is chosen as the D-measure for assessing direct discrimination: the confidence of the rule with PD item(s) A, divided by the confidence of the rule whose premise contains $\neg A$. Extended-lift-lower-bound (*elb*) is chosen as D-measure for assessing indirect discrimination, which adapts *elift*, a variation of *slift*, with the confidence information from the background knowledge (Ruggieri et al., 2010a).

We combine this with established distribution-based measures of item(set) interestingness: Based on the *D-measure*, we define the item-focused *AD-measure* of the accumulated degree of discrimination of an individual item, and Mutual Information as a measure of correlation between pairs of items, based on the discriminatory rules these items are involved in. We also combine the rule-focused *D-measure* with a similarity measure on rule pairs to relate several rules to one another in their degree of discrimination.

Item Focusing: The goal of item focusing is to determine the degree of discrimination in individual items and to be able to relate these to one another in order to gain a meta-level view.

A simple measure is the *supp (support)* of an item or items in the mined rule set. Let an item be characterized as the equality of an attribute q to a value (range) v, let N be the number of rules, and $N(x)$ the number of rules satisfying the argument x. Let $Q = V$ denote an itemset: $q_1 = v_1, ..., q_m = v_m$ for $m \geq 1$. Then

$$supp(Q = V) = \frac{N(Q = V)}{N} \tag{1}$$

This measure does not take into account how discriminatory the PD rules are; therefore we define the *AD-measure* as a form of averaged *support*, which is weighted by the D-measure of these rules. Let R_i be the left-hand side of rule i and $D\text{-}measure_i$ its D-measure, then:

$$AD\text{-}measure(Q = V) = \frac{\sum_{i=1}^{N} D\text{-}measure_i \cdot b(Q = V, R_i)}{supp(Q = V)} \quad (2)$$

$$\text{with } b(Q = V, R_i) = \{ \begin{array}{ll} 1 & if\ R_i\ contains\ Q = V \\ 0 & else \end{array}$$

The AD-measure's range is $[0, \infty)$. (In the visualizations, we only show the AD-measure for single items.)

Other item-focusing interestingness measures have been proposed such as the magnitude-based and association-based interestingness functions in (Bhandari, 1994) and the summary-based interestingness functions in (Hilderman et al., 1999), which are attribute-oriented – the probability distribution of all the items within one attribute is taken into account. However, the attribute-oriented measures are limited in the sense that although the associations between two or more attributes (e.g. $foreign\_worker$ and *purpose*) can be well captured and ranked, the association between two or more items is ill-measured (e.g. $foreign\_worker = yes$ and $purpose = new\_car$). When an attribute only contains one value in all records, as in most of the rules extracted by DCUBE, the attribute-oriented measures would take it as a non-interesting item (see section 2.2.3 in (Bhandari, 1994)). In such a case, the items need to be treated individually, in other words, each item is taken as a non-dividable entity without the context of attributes, i.e. item-focusing. The support measure is in accord with this criterion, and so is *Mutual Information (MI)* (Sy, 2003) on pairs of items in the rules. We can use MI to characterize the interdependency between two items:

$$MI(q_1 = v_1, q_2 = v_2) = \log \frac{p(q_1 = v_1, q_2 = v_2)}{p(q_1 = v_1) \cdot p(q_2 = v_2)} \quad (3)$$

with $p(Q = V) = N(Q = V)/M$, with M the total number of items in all rules (including overlap).

Interestingness measures can be categorized into objective and subjective measures. Objective measures such as support, confidence, mutual information and lift measures focus on the probability distribution of data and the predictive accuracy of rules. Subjective measures take a user's judgement and experience as a part of the measurements, such as exceptions (Gonçalves et al., 2005) and expectations (Liu et al., 1999), which involve a user's predefinition of a specific set of items. The "Discrimination Discovery in databases" approach also lets a user subjectively select PD items. Thus, *AD-measure* is a hybrid objective/subjective measure.

Rule Focusing: The support-confidence framework can be insufficient to evaluate the quality of a rule (Bayardo & Agrawal, 1999). Many rule-focusing measures of interestingness have been proposed, including objective measures such as rule-interest, lift, conviction, Loevinger index, implication intensity, coverage, strength, performance, Sebag and Schoenauer index and IPEE, and subjective measures such as simplicity, unexpectedness and actionability (Liu et al., 1999; Blanchard et al., 2007a). The D-measures in DCUBE system also contain subjectiveness in that they take the user's prior knowledge of the data domain into account. We will use the D-measures (*slift* and *elb*) to characterize each rule's interestingness.

To characterize the similarity/distance between two rules, we concentrate on all the items in the rule (for association rules) or all items on the left-hand side of the rule (for classification rules) and regard them as an itemset or vector in the space spanned by all attributes. We can then apply a number of measures such as the Hamming distance, Jaccard distance, Dice's coefficient, or Cosine similarity and its binary version, the Tanimoto coefficient. Another measure of the association between two (or more) rules is through fuzzy meta-association rule extraction (Yahia & Nguifo, 2004). We will use the Jaccard distance to indicate the degree of dissimilarity between two rules. In contrast to other measures, it isapplicable in a straightforward way to sets with different items as well as readily implemented. due to its simplicity, applicability to sets with different nu Let $S(R)$ denote the itemset of R. Then the Jaccard distance between R_1 and R_2 is $J(R_1, R_2) = 1 - (S(R_1) \cap S(R_2)/S(R_1) \cup S(R_2))$.

4. DCUBE-GUI

DCUBE-GUI, our tool for visualizing the degree of discrimination inherent in rules and items, is built in Java SE6, mainly with Processing-1.2.1) for graphical design[6]. It also uses packages from Weka-3.6.4[7] and Jtreemap(1.1.0)[8].

The PD rules in DCUBE can be directly extracted from the database via a JDBC connection in the tool, or we can extract the rules in advance, which are then stored in on the local machine for the tool to read[9]. The latter is recommended because querying the database can be time-consuming, which inhibits repetitive querying. In order to accurately reflect the general information "trends" or distribution of the

[6] http://processing.org/

[7] http://www.cs.waikato.ac.nz/ml/weka/

[8] http://jtreemap.sourceforge.net/

[9] In our demonstration, two .csv files of rules are used

rules from a high level, we need not only a large quantity of rules, but also good quality (highly discriminatory). Based on these two criteria, two sets of PD rules are extracted from DCUBE for visualization. On the German Credit Dataset, this led to one set of 1062 rules extracted in DD analysis, with minimum *support* > 20 and *slift* > 2.6, we call it Rule Set 1 (RS1). The other set contains 770 rules extracted in IDD analysis, with minimum *support* > 20 and *elb* > 1.3, we call it Rule Set 2 (RS2). The visualizations on the two rule sets are twofold.

Item Oriented: Figure 2 shows the interface of the tool. On the left is a panel for item information. The lengths of the bars are scaled and sorted according to the supports of the items. In the middle is the view of bubbles, each bubble represents an item, of which the size is scaled according to its support. In the middle, colour indicates PD (red) and PND (blue); on the right of the figure, rainbow colors indicate the AD-measure. The rainbow-colors scale ranges from red (high) via orange, yellow, green, and blue to purple (low). From Figure. 2, we get an overview of the distribution of the items in RS1. The items *foreign_worker* = *yes* and *personal_status* = *female_div_or_sep_or_mar* are very frequent, because almost every rule contains at least one the two PD items. We also see that the PD item *age* = *gt_52d6* is not frequent, which implies that the assumption that old people are often discriminated in various situations is not true. However, besides the two PD items, we also see other major items in the RS1, such as *other_payment_plans* = *none*, *credit_amount* = *le_38848d8*, *own_telephone* = *none*, etc. On the right, we see that although items such as *foreign_worker* = *yes* and *personal_status* = *female_or_div_or_sep_or_mar* are larger in size, their AD scores are not high (yellow or green) compared to items *employment* = *lt_1*, *own_telephone* = *none* and *purpose* = *new_car* (red or orange). In terms of AD-measure, the red or orange items appear more "effectively" in the rules than the yellow or green ones. Especially for *employment* = *lt_1*, the item appears not frequently in general, but quite frequently in the rules with high discriminatory scores, which shows that a person in a disadvantaged group who has less than one year working experience is often rejected by the bank on a loan, hence, discriminated.

The relationships between items are explored next. In Figure. 3 (a), the bubbles are aligned and connected with semi-circles, of which the weights are determined according to the pairwise Mutual Information (MI) between items. The threshold of MI in Figure. 3 (a) is 0.06 (half the maximum MI value). We see among all the associations above the given threshold, the item *foreign_worker* = *yes* is strongly related to *own_telephone* = *none* and *credit_amount* = *le_38848d8*, which means the two combinations appear more often and are more influential than the others. We can interpret this as follows: a foreign worker is often discriminated when (s)he does not own a telephone and/or has less than 38848.8 units of credit. We also discover that *personal_status* = *female_div_or_sep_or_mar* has a strong connection with *property_magnitude* = *real_estate* and *age* = *le_30d2*. This indicates that non-single women tend to be discriminated against when under 30 years old or when owning real estate, which is a surprising discovery.

An alternative visual presentation of the association between items is shown in Figure. 3(b). This time we investigate the ruleset in RS2. Each arc in the circle represents an item. The weights of the arcs are scaled according to the counts of items. The white curves connecting different arcs represent associations between items, and their weights are also based on MI, as the semi-circles in Figure 4. Thicker connection means stronger association. When the user hovers the mouse over a certain item, the relevant connections will be shown, the others are hidden. As we observe through visualizations of the rules extracted from indirectly discriminatory analysis (which performs background checking), new knowledge is discovered: we find new strong links between *personal_status*=*female_div_or_sep_or_mar* and two PND items: *housing* = *rent* and *employment* = *lt_1*. This indicates that a non-single woman tends to be discriminated against if currently renting a house or having an employment experience of less than one year.

Rule Oriented: Rule-Oriented visualization shows an overview as well as a general distribution of all the PD rules, how some (combinations of) items affect this distribution, and whether there is an underlying pattern with respect to D-measures.

With the Jaccard distance, hierarchical clusters of rules are built, with each rule as a leaf in a dendrogram. We use agglomerative hierarchical clustering (Fisher, 1987), with the linkage criterion Weighted Pair Group Method of Arithmetic Average.

Then, we use the Squarified Treemap (Johnson & Shneiderman, 1991; Bruls et al., 1999) space filling approach to visualize the dendrogram of rules, with each elementary rectangle as a PD rule. An elementary rectangle is weighted as well as colored with rainbow colors based on the D-measures, so that a rule with higher discriminatory score is more reddish, the one with lower score is more purplish. Figure 5 shows

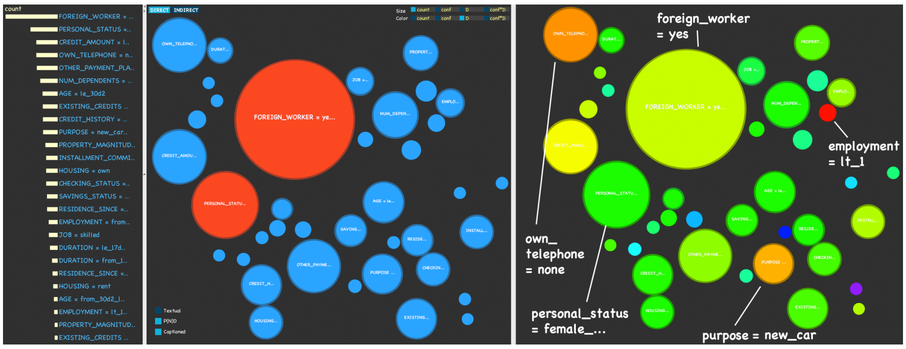


Figure 2. Visualization with bubbles on RS1.

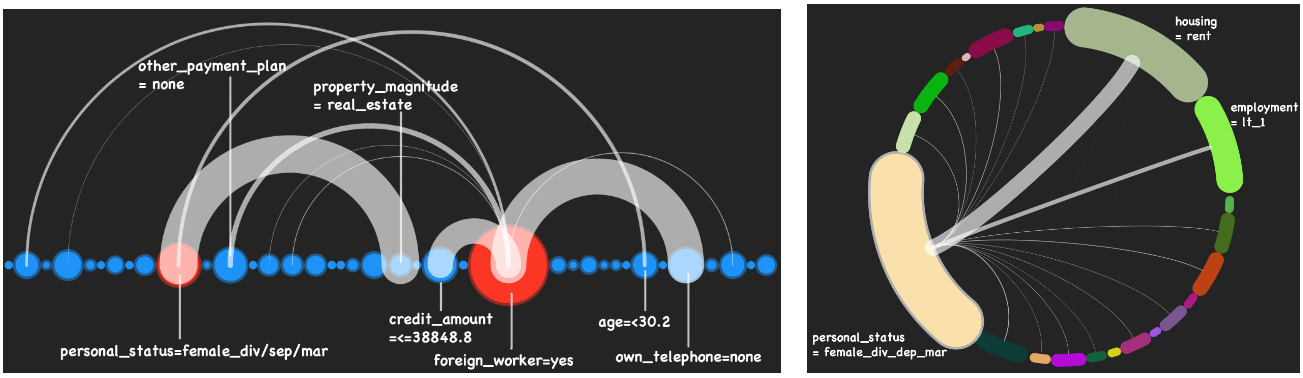


Figure 3. Associations between items (a, left) in Bubble View on RS1; (b, right) in Arc View on RS2.

a treemap visualization on the PD rules extracted in the IDD analysis. We see that on the left, the rules are clearly clustered into distinct regions and the D-measure distribution is strongly correlated with the distribution of the regions (since we color the rectangles based on the rules' *elb* scores), which implies that certain combinations of items within the rules lead to more discriminatory situations than others. E.g. the region at the bottom starting from the left (red) and the region at the top-left corner (green) contain rules with a relatively high D-measure, which should inspire further investigation.

The items are aligned on the right hand side of Figure. 4 (right), sized according to their supports. User can select multiple items at a time; the rules (rectangles) not containing the selected items will be colored white. We can see that the remaining colored areas contain the items *housing* = *rent* and *own_telephone* = *none*, which means this combination of these items quite often appear in high-ranked PD rules.

5. Conclusions and future work

5.1. Summary

We have proposed new interestingness measures and a visualization approach, that aids users in analyzing higher-level information such as the relative importance of items and the relationship between items and between rules. We have used discrimination-aware data mining (Pedreschi et al., 2008; Ruggieri et al., 2010a) to illustrate the advantages of this approach, building a tool that imports classification rules obtained by SQL querying in the DCUBE tool (http://kdd.di.unipi.it/dcube). While this approach is generally applicable, we have argued it is particularly interesting for discrimination-aware and privacy-aware data analyses, since it can integrate scores that measure, e.g. the discriminatory or – by extension – privacy-violating potential of a rule, thus enhancing explorations in these open-ended fields in which the semantics of what is discriminatory or privacy-violating keep changing. This is only a first step in developing rich and effective visualizations for these fields (most notably, the problem of how to evaluate the proposals, which is well-known in visual data mining, applies here as well). Yet, we believe that the approach not only supports the specific task chosen for demonstration, but also extends to many other areas. We will sketch some of these in the remainder of this section.

5.2. Visualizing other types of patterns

DCUBE-GUI can be directly applied to the analysis of arbitrary classification rules; the D-measure weights can be set to 1 or replaced by another interestingness measure, depending on the application. Different gen-

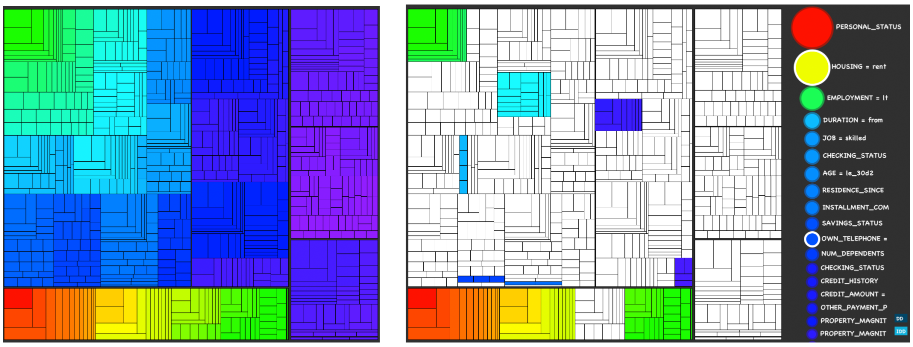


Figure 4. Treemap Visualization on RS.

eralizations for association-rule visualization are possible; a straightforward form would be to extend the considered itemsets to also contain the rule's right-hand side. DCUBE-GUI could also be applied to other classifiers such as decision trees, by transforming the tree to rules. This could be applied, for example, to the following application.

5.3. Feedback and awareness tools

DCUBE-GUI was designed also to be a a first step towards the next generation of feedback and awareness tools for privacy protection. The general idea of feedback and awareness tools (e.g. for privacy: (Lederer et al., 2004; Liu et al., 2006; Nguyen & Mynatt, 2002; Hansen, 2008)) is to play selected parts of a user's actions and their possible/likely consequences back to the user in order to create more awareness and reflection of their actions.

A recent proposal is the Privacy Wizard by (Fang & LeFevre, 2010). The idea is to support users in choosing their privacy settings in online social networks such as "with whom to share one's birthday": From a training set of positive and negative examples elicited from the user with active learning, the system learns a classifier that predicts from new users' profile and relationship features whether the current user will want to share her birthday with them. The result can be proposed by a wizard for a new "friend", or the learned decision tree can be shown (in expert mode; a standard decision-tree image).

A wizard is "a user interface type that presents a user with a sequence of dialog boxes that lead the user through a series of well-defined steps" (Wikipedia, 2011). In line with this, the goal of the Privacy Wizard is to automatically configure the user's future privacy settings (because the tool has learned "types of friends" – for a new contact, the system could propose the settings). The assumption behind this is a well-defined notion of privacy: *privacy as hiding* and access control as the means to reach it (*privacy as control*) (Gürses & Berendt, 2010). However, the system would take any choices the user made and apply them. Thus, for example, it could find that "I don't want people who eat beans to know my birth date", and would offer to automatically apply this to future contacts.

However, in a field such as privacy neither the task nor the steps to reach it may be so well-defined. Extending the fictitious example, we might find that in fact all people who eat beans in the existing list of friends are migrants – thus, choices made on the basis of bean-eating could be regarded as racist. These findings could be based on the visual inspection of items or of the similarity of items or rules. In the context of one's private social network, such a "background" may or may not in fact exist, and a user may or may not decide to reconsider their choices of information disclosure, but our tool would make this possibility visible and thus a possible object of awareness and reflection. (In the context of granting bank loans under certain jurisdictions, the situation is different: the criterion of country of origin *must* not be used as basis for a decision.) Once such reflection is started, it can lead to further reflections, for example on the consequences of disclosing one's favourite foods in social networks, and thus support the continuous (re-)negotiations of what is public and what is private that underlie the wider notion of *privacy as practice* (Gürses & Berendt, 2010). Privacy is not static and evolves in a dynamic social, technological and legal environment. Therefore, tools that allow for more open-ended definitions of the constructs and goals at hand (such as privacy) and of the actions appropriate for reaching them (such as privacy-protecting actions) are likely to be more helpful in such areas, increasing awareness and active participation in the protection of one's own and other's private spheres.

In future work, we will take these ideas further. In particular, the set of analyzable patterns will be extended, an interface will be provided to deploy interestingness measures in a modular way, and the requirements and experiences of end users will be collected and evaluated to validate measures and tool features.

Acknowledgements. We thank Salvatore Ruggieri and Dino Pedreschi for discussions and comments during the development of DCUBE-GUI.

References

Bayardo, Jr., R. J., & Agrawal, R. (1999). Mining the most interesting rules. *Proc. KDD'99* (pp. 145–154). New York, NY, USA: ACM.

Berendt, B., Preibusch, S., & Teltzrow, M. (2008). A privacy-protecting business-analytics service for online transactions. *International Journal of Electronic Commerce*, *12*, 115–150.

Bhandari, I. (1994). Attribute focusing: machine-assisted knowledge discovery applied to software production process control. *Knowl. Acquis.*, *6*, 271–294.

Blanchard, J., Guillet, F., & Briand, H. (2007a). Interactive visual exploration of association rules with rule-focusing methodology. *Knowl. Inf. Syst.*, *13*, 43–75.

Blanchard, J., Pinaud, B., Kuntz, P., & Guillet, F. (2007b). A 2D-3D visualization support for human-centered rule mining. *Computers Graphics*, *31*, 350 – 360.

Bruls, M., Huizing, K., & van Wijk, J. (1999). Squarified treemaps. *Proc. Joint Eurographics and IEEE TCVG Symposium on Visualization* (pp. 33–42). Press.

Calders, T., & Verwer, S. (2010). Three naive Bayes approaches for discrimination-free classification. *Data Min. Knowl. Discov.*, *21*, 277–292.

Fang, L., & LeFevre, K. (2010). Privacy wizards for social networking sites. *Proc. WWW 2010.*

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Mach. Learn.*, *2*, 139–172.

Frank, A., & Asuncion, A. (2010). UCI machine learning repository. `http://archive.ics.uci.edu/ml`.

Gonçalves, E., Mendes, I., & Plastino, A. (2005). Mining exceptions in databases. In *AI 2004*, LNCS 3339, 1–29. Springer, Berlin / Heidelberg.

Gürses, S., & Berendt, B. (2010). The social web and privacy: Practices, reciprocity and conflict detection in social networks. In E. Ferrari and F. Bonchi (Eds.), *Privacy-aware knowledge discovery: Novel applications and new techniques.* Chapman & Hall/CRC Press.

Hajian, S., Domingo-Ferrer, J., & Martínez-Ballesté, A. (2011). Discrimination prevention in data mining for intrusion and crime detection. *IEEE SSCI 2011.*

Hansen, M. (2008). Linkage control - integrating the essence of privacy protection into identity management. *eChallenges.*

Hilderman, R. J., Hamilton, H. J., & Barber, B. (1999). Ranking the interestingness of summaries from data mining systems. *Proc. Twelfth International Florida Artificial Intelligence Research Society Conference* (pp. 100–106). AAAI Press.

Johnson, B., & Shneiderman, B. (1991). Tree-maps: a space-filling approach to the visualization of hierarchical information structures. *Proc. 2nd conference on Visualization '91* (pp. 284–291). Los Alamitos, CA, USA: IEEE Computer Society Press.

Kuntz, P., Guillet, F., Lehn, R., & Briand, H. (2000). A user-driven process for mining association rules. *Proc. ECML/PKDD'00* (pp. 483–489).

Lederer, S., Hong, J. I., Dey, A. K., & Landay, J. A. (2004). Personal privacy through understanding and personal privacy through understanding and action: Five pitfalls for designers. *Personal and Ubiquitous Computing*, *8*, 440–454.

Liu, B., Hsu, W., Mun, L.-F., & Lee, H.-Y. (1999). Finding interesting patterns using user expectations. *IEEE Transact. Knowl. and Data Engineering*, *11*, 817 –832.

Liu, H., Maes, P., & Davenport, G. (2006). Unraveling the taste fabric of social networks. *International Journal on Semantic Web and Information Systems*, *2*, 42–71.

Nguyen, D., & Mynatt, E. (2002). *Privacy mirrors: Understanding and shaping socio-technical ubiquitous computing* (Techn. Report). Georgia Institute of Technology.

Pedreschi, D., Ruggieri, S., & Turini, F. (2008). Discrimination-aware data mining. *Proc. KDD'08* (pp. 560–568). ACM.

Pedreschi, D., Ruggieri, S., & Turini, F. (2009). Measuring discrimination in socially-sensitive decision records. *SDM* (pp. 581–592).

Ruggieri, S., Pedreschi, D., & Turini, F. (2010a). Data mining for discrimination discovery. *TKDD: ACM Transactions on Knowledge Discovery*, *4*.

Ruggieri, S., Pedreschi, D., & Turini, F. (2010b). DCUBE: discrimination discovery in databases. *Proc. SIGMOD'10* (pp. 1127–1130).

Sy, B. K. (2003). Discovering association patterns based on mutual information. *Proc. 3rd international conference on Machine learning and data mining in pattern recognition* (pp. 369–378). Berlin, Heidelberg: Springer-Verlag.

Wikipedia (2011). Wizard (software) — wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Wizard_(software)&oldid=418471149`.

Wong, P. C., Whitney, P., & Thomas, J. (1999). Visualizing association rules for text mining. *Proc. 1999 IEEE Symposium on Information Visualization* (pp. 120–). Washington, DC, USA: IEEE Computer Society.

Yahia, S. B., & Nguifo, E. M. (2004). Emulating a cooperative behavior in a generic association rule visualization tool. *Proc. 16th IEEE International Conf. on Tools with Artificial Intelligence* (pp. 148–155). IEEE.

Influence of Size on Pattern-based Sequence Classification

Menno van Zaanen MVZAANEN@UVT.NL
Tanja Gaustad T.GAUSTAD@UVT.NL
Jeanou Feijen J.H.M.FEIJEN@UVT.NL
Tilburg University, PO Box 90153, 5000LE, Tilburg, The Netherlands

Abstract

In this paper we investigate the impact of size of vocabulary, the number of classes in the classification task and the length of patterns in a pattern-based sequence classification approach. So far, the approach has been applied successfully to datasets classifying into two or four classes. We now show results on six different classification tasks ranging from five to fifty classes. In addition, classification results on three different encodings of the data, leading to different vocabulary sizes, are explored. The system in general clearly outperforms the baseline. The encodings with a larger vocabulary size outperform those with smaller vocabularies. When using fixed length patterns the results are better, but vary more compared to using a range of pattern lengths. The number of classes in the classification task does not have an effect on the trends in the results, although, as could be expected, tasks with fewer classes typically lead to higher accuracies.

1. Introduction

The area of music information retrieval (MIR) aims at extracting information from music (Downie, 2003). This includes tasks such as the automatic analysis of music, query by humming, but also the design and implementation of music archives and collections. In particular, we are interested in the sub-area of computational methods for the classification of music.

Classification of music is typically performed on datasets that contain musical pieces grouped in classes. In the most well-known tasks, pieces are grouped by genre, composer, or musical period (see e.g. (Basili et al., 2004; Backer & van Kranenburg, 2005; Kyradis, 2006; Geertzen & van Zaanen, 2008)). This comes down to designing and building a system that identifies the genre, composer, or musical period of a particular piece. Such a system allows users to search their music collection through additional views based on the meta data that is automatically assigned to the pieces.

For the classification of musical pieces, a supervised machine learning approach is commonly applied. Given a set of training data consisting of musical pieces with corresponding class information, a classification model is learned that describes how features are related to classes. The features are used to allow for a standardized and compact representation of important aspects of the musical pieces.

Features are typically divided into two groups. *Local features* describe aspects of the piece by looking at small parts of the piece, i.e. events in short time intervals (e.g. tempo changes). *Global features* encompass measures describing aspects of entire musical pieces, such as meter, distributions of intervals, or key.

In this paper, we introduce a classification system that analyzes local features from a symbolic representation of musical pieces and finds patterns that help with classification. The approach will be tested on six different classification tasks: distinguishing music by composer (two datasets), by country of origin (two datasets), by musical period and by the initial letter of the last name of the composers. Our approach is based on identifying patterns of a particular length and weighing them according to their relative importance.

The described feature extraction and classification approach is essentially task independent. It can be applied in situations where instances have an inherent sequential nature. Classification of unseen data is based on patterns found in the instances in the training data, which means that the approach relies on the ordering of the symbols within the instances.

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands*.

Even though the system is generic, we will illustrate the approach in the context of MIR as we can represent music in alternative ways using different encodings. Each encoding has its own properties. This allows us to investigate the relationship between said properties and the performance of the system.

The paper is structured as follows. In section 2, we describe our classification approach in more detail. The datasets that are used in the experiments, including a description of the classes and features, together with their related statistics are elaborated on in section 3. Our results are presented and discussed in section 4. Section 5 concludes the paper with a summary of our major findings and an outlook on future work.

2. Pattern-based sequence classification

The classification system (van Zaanen & Gaustad, 2010) aims at identifying patterns that can be used to classify (new, unseen) sequences. These patterns are in the shape of subsequences, i.e. consecutive symbols, and are extracted from the sequences in the training dataset. For practical purposes, patterns consist of subsequences of a pre-determined, fixed length. This means that they can be seen as n-grams, where n defines the length of the pattern (Heaps, 1978). The system only retains and uses patterns that are deemed useful according to some "usefulness" or scoring metric (described in section 2.1).

The underlying idea behind this approach is that we are aiming to identify interesting patterns for a particular class with respect to the other classes the system needs to distinguish. This is similar to an approach that learns a language model for each class, but in contrast, our approach takes the languages of the other classes into account as well. Effectively, the system learns boundaries between languages (which are represented by different classes).

By using n-grams as the representation of our patterns, we explicitly limit the class of languages we can identify. In fact, using patterns of a specific length, we can learn the boundaries of languages taken from the family of k-testable languages (Garcia & Vidal, 1990). By definition, this family contains all languages that can be described by a finite set of subsequences of length k. The k corresponds exactly with the length of the n-gram patterns. This also means that if we need to be able to identify information in sequences that span over symbols of length $> n$ (a longer distance dependency), we cannot expect the system to distinguish the correct class.

2.1. Scoring metric

During training we would like to identify patterns (i.e. n-gram subsequences) that are maximally discriminative between languages (or classes). Additionally, we would like to find patterns that occur frequently, since that increases the likelihood of the patterns occurring in the test data. Both of these properties are contained in the scoring metric that is used here.

To measure the effectiveness and usability of the patterns, we apply a classic statistical measure from the field of information retrieval, namely the "term frequency*inverse document frequency" (*tf*idf*) weight (van Rijsbergen, 1979). This measure consists of two components: term frequency (*tf*) which measures the regularity and inverse document frequency (*idf*) which measures the discriminative power of the pattern. Combined, they provide a measure that gives high weight to patterns that have a high discriminative power and, at the same time, occur frequently. Assuming that the testing data is similar to the training data, and given the fact that these patterns both occur frequently in the training data and also have a high discriminative power, this leads us to conclude that these patterns will be helpful for classification.

Originally, in the context of information retrieval, the *tf*idf* weight is used to evaluate how relevant a document in a large document collection is given a search term. In its classic application, *tf*idf* weights are computed for all documents separately in the collection with respect to a search term.

We, however, use the term "document" in a different way. Typically, in the field of information retrieval, all musical pieces would represent separate documents, but here we are interested in the usefulness of a pattern with respect to a particular class. Therefore, we take as "document" all musical pieces belonging to a particular class combined. This allows us to measure the relevance of a pattern with respect to all musical pieces that belong to the same class in one go. Effectively, we will have as many documents as there are classes in the dataset.

The first component of the *tf*idf* metric is the *tf*, which is defined as the number of occurrences of a given term in a document. Counting the number of occurrences of a term and comparing it against the counts in other documents results in a preference for longer documents (which overall contain more terms). To reduce this effect, counts are normalized by the length of the document, leading to Equation 1.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

$n_{i,j}$ denotes the number of occurrences of term t_i in document d_j. The denominator represents the length of document d_j, which is measured as the total number of terms in document d_j.

Scoring patterns based on *tf* already provides interesting information. Assume that we are trying to identify documents that are relevant to a particular search term. If a document has many occurrences of the term, it is quite likely to be about that topic. On the other hand, if a document does not contain the term at all ($tf = 0$) then that document is probably not about this term. Obviously, the document may still be about the same topic, but might use different words, such as synonyms, for the topic. In fact, this has led to research into, for example, lemmatizing, stemming, pseudo relevance feedback and automatic synonym generation (Baeza-Yates & Ribeiro-Neto, 1999).

Another problem with simply using the *tf* scoring metric is that there are terms that occur very frequently in all or at least most documents. For instance, function words, such as *the*, *a*, and *and*, probably occur in all documents. When a function word is used as a search term, documents with many function words are considered very relevant (since the *tf* will be very high). However, the weight given to function words will also be higher than that of content words.

To reduce the effect of the function words, or often occurring terms that appear in all documents, the *tf* metric is extended with an *idf* component as described in Equation 2.

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|} \tag{2}$$

$|D|$ is the total number of documents in the collection and $|\{d : t_i \in d\}|$ is the number of documents that contain the term t_i.

The *idf* metric measures relevance of a term with respect to all documents. In other words, terms that occur in all documents are less useful when deciding document relevance compared against terms that occur in fewer documents. When relevance is computed based on a set of terms the *idf* gives higher weight to terms that only occur in fewer documents.

To obtain the *tf*idf* weight for a particular term, the term frequency *tf* and inverse document frequency *idf* are combined as shown in Equation 3.

$$tf{*}idf_{i,j} = tf_{i,j} \times idf_i \tag{3}$$

The default way of computing *tf*idf* provides us with an indication of how relevant a particular document is to a particular term. This metric can be extended to multiple search terms. Once the *tf*idf* for all documents is computed for each term, the *tf*idf* values are summed and the most relevant document, i.e. with the highest *tf*idf* score, is selected.

Here, we extend the *tf*idf* metric differently. In addition to combining the *tf*idf* values of a set of terms, we compute *tf*idf* values of sequences of terms thereby preserving the underlying order. This amounts to using n-grams (with $n \geq 1$) as terms. Our intuition is that subsequences are more informative than single terms to determine boundaries between languages.

The modification of the computation of the *tf*idf* weights is rather straightforward. Instead of counting single terms, n-grams are counted as if they are single terms (with single terms being a specific case where $n = 1$). For instance, $n_{i,j}$ is the number of occurrences of a particular n-gram t_i in document d_j.

2.2. System walk-through

During training, the system receives a collection of classes, each consisting of a set of sequences that come from the underlying language of that class. Based on these sequences, all possible n-grams are extracted and for each of these patterns the *tf*idf* score is calculated. This leads to a set of patterns combined with a score for each class. A score signifies how well the pattern fits the particular class (the *tf* of the class) and is weighted by the *idf* (which indicates uniqueness of the pattern with respect to the other classes). Patterns that have a *tf*idf* score of zero are removed, as they are useless for classification purposes. This may happen when the pattern occurs in all classes.

After the training phase, the system receives a new sequence which needs to be classified. Based on the sequence, the system computes an overall score for each class. This is done by applying each pattern to the sequence and for each match, the *tf*idf* scores for that pattern are added to the cumulative scores for each class. Finally, the system selects the class that has the highest score as the “correct” class for the sequence.

We have described a system that uses fixed length patterns, but it is also possible to use patterns of varying length. This is arranged by computing scores of a sequence using patterns of different length. The final score of the sequence is the linear combination of the scores of each pattern length.

Taking another look at using long and short patterns simultaneously, we see that shorter patterns typically have a higher *tf*idf* score. This is due to the fact that shorter patterns have a higher overall likelihood of occurring in a sequence than longer patterns. Since the number of occurrences is taken into account (in the *tf* component of the *tf*idf* score), this means that shorter patterns typically have a higher score than longer patterns. On the other hand, if a long pattern, which is more precise, is found in a sequence during classification, it should probably have more influence in the overall decision. We have experimented with (multiplicative) length normalization of the patterns, but this did not have any significant effect.

3. Data

3.1. Dataset and classification tasks

To analyze and evaluate our approach, we retrieved data that can be found under the "composers" section on the front page of the **kern scores website[1] (Sapp, 2005). This data contains symbolic representations of musical pieces from 50 composers. Using these pieces, we compiled six different classification tasks. All tasks use pieces from this one dataset. Table 1 provides an overview of the six tasks including the number of classes and number of musical pieces.

Table 1. Overview of the six classification tasks.

Dataset	# classes	# pieces
composers	50	3132
composers50	9	2747
countries	14	3132
countries50	5	3050
periods	6	3132
initials	17	3132

The first two tasks involve classifying the musical pieces according to their composer. The first dataset consists of the pieces of all 50 composers (*composers*). However, we found that there are several classes (composers) that only have a small number of pieces in the dataset. To remove some of the skewedness, we created a second dataset by imposing a frequency threshold and retaining only those composers that have 50 or more musical pieces in the original set (*composers50*).

The next two tasks aim at classifying the pieces according to the country of origin of its composer. Again, two datasets are created, on the one hand using all data (*countries*) and on the other hand, similarly to the division in the composers tasks, restricting the classes to countries with at least 50 pieces (*countries50*).

The fifth task involves classification into musical periods (*periods*). Each piece is assigned to one of six periods: Middle Ages (400–1400), Renaissance (1400–1600), Baroque (1600–1750), Classical (1750–1800), Romantic (1800–1900), Modern (1900–today). The information required for assigning countries and musical periods of composers was taken from the International Music Score Library Project (IMSLP) website[2].

The final task has been setup as a sanity check: The classes contain pieces of composers that share the first letter of the last name (*initials*). Since the system aims at learning regularities within (and between) classes and since we do not assume any real regularities within pieces of these artificial classes, we expect the performance here to be lower than that of other tasks.

3.2. Data representation

Starting with a set of musical pieces in **kern format (Huron, 1997), the music is transformed into different symbolic representations. The **kern format is an ASCII representation of sheet music. Notes (and other musical symbols) are grouped by voice, which allows us to extract notes of a particular voice. The experiments described here are all use the notes contained in the first voice in the **kern files. By applying a transformation on the notes, we can select which properties of the data we would like to use while working with the same underlying data.

There are two musical aspects we extract information from: pitch and duration. For both aspects, three different encodings are used: *absolute*, *relative*, and *contour*. In this paper, the same encoding is used for both pitch and duration. Note that many more encodings and combinations of encodings are possible, but since that will explode the experimental space we only focus on these three encodings here.

The *absolute* encoding for pitch refers to the absolute value in semitones (e.g. $c = 0$, $d = 2$, $e = 4$, $c' = 12$, etc.). For duration a similar encoding is used where each note length has its own symbol. The *relative* encoding describes pitch changes with respect to the previous note in the number of semitones and direction of change (e.g. 4, -7). A similar encoding is designed for duration, which describes the relative duration of consecutive notes. Finally, *contour* only models whether the pitch or duration rises (1), falls (-1) or stays the same (0). Contour encoding is also known as the Parsons Code (for Melodic Contours) (Parsons, 1975).

The major difference between the various encodings is that the number of unique symbols (i.e. the size of the vocabulary or the number of types) is significantly different. The classification tasks do not have any influence on the number of unique symbols. Given

[1] http://kern.ccarh.org/

[2] http://www.imslp.org/

Table 2. Mean number of unique symbols per encoding and their standard deviation.

Encoding	Mean # of unique symbols
Relative	1075 (±75.69)
Absolute	727 (±57.75)
Contour	12 (±00.00)

the same encoding, the size of the vocabulary will remain constant across different tasks. An overview of the mean number of unique symbols per encoding and their standard deviations can be found in Table 2.

Another difference between the encodings is the amount of information described by the symbols. The contour encoding only contains information on the direction, whereas the relative encoding describes the interval. However, it does not have information on the exact notes being used. The absolute encoding denotes exact notes, so all information is encoded. However, it may be argued that the relative encoding fits better with how patterns in music are identified. Sequences of notes are recognized as similar even when the sequence starts on different notes.

4. Results

When applying[3] the system to the datasets based on the different encodings there are many aspects that can be investigated. Firstly, we will look at the number of patterns that are generated. In section 3.2, we already saw that the classification task does not have a significant impact on the number of unique symbols. Also, the classification task does not have a significant impact on the number of patterns, so we only show this information per encoding and n-gram size. As can be seen in Table 3, for all encodings the number of useful patterns (based on the *tf*idf*) grows when the patterns get longer. This is due to the fact that longer patterns have a lower likelihood of occurring in all classes (which would mean that the *idf* is zero). Furthermore, the contour encoding finds fewer useful patterns. This is because the number of unique symbols is much smaller (see Table 2), which means that it is more likely that the patterns occur in all classes.

Comparing the changes in number of patterns between relative and absolute encoding, the following picture emerges. At first, larger vocabulary sizes (relative encoding) lead to more possible patterns as can be seen when the pattern is of length one. However, when the patterns get longer, the absolute encoding leads to more patterns. This is because the number of (combinations of) intervals per class is more likely to occur within different classes than the exact combination of notes. In other words, the relative encoding, which describes intervals, generalizes over the absolute encoding, which describes exact notes.

Table 3. Mean number of useful patterns per encoding and pattern length.

	Pattern length			
	1	2	3	4
Relative	2005	25275	83291	147305
Absolute	1047	27018	108467	189957
Contour	4	111	1668	12079

Next, we look at the accuracy results for experiments where all patterns have the same length (fixed length n-grams). These results are depicted in Figure 1.

The first thing to notice is that the systems overall significantly outperform the majority class baseline (although in some cases the accuracies are lower than the baseline). In general, longer patterns perform worse, which is expected as long patterns are not likely to reoccur in the test data. Similarly, very short patterns (n=1) do not perform well, as the patterns that are retained (often accidentally) do not occur in all classes. In this case, the short patterns give a false sense of predictive power. If no patterns match the test sequence, the system falls back on the majority class baseline.

Considering the different encodings, it is clear that the contour representation performs worst. However, results get better with longer n-grams. This can be explained by the more slowly growing pattern space (see Table 3). The absolute and relative encodings perform similarly (there is no statistically significant difference between absolute and relative encodings, but the contour encoding performs significantly worse) and both peak at $n = 3$ or 4 depending on the classification task.

Comparing the classification tasks with and without frequency cutoff (*composers* vs. *composers50* and *countries* vs. *countries50*), we can observe an increase in accuracy for the two tasks with a frequency cutoff. The most probable explanation is the reduction of classes involved. Keep in mind that the classes that have a small number of pieces are removed in the frequency cutoff datasets.

Interestingly enough, our classification system produces results that lie above the majority class baseline for the *initials* task albeit the accuracy is overall

[3] All results presented in this paper are computed using 10-fold cross validation.

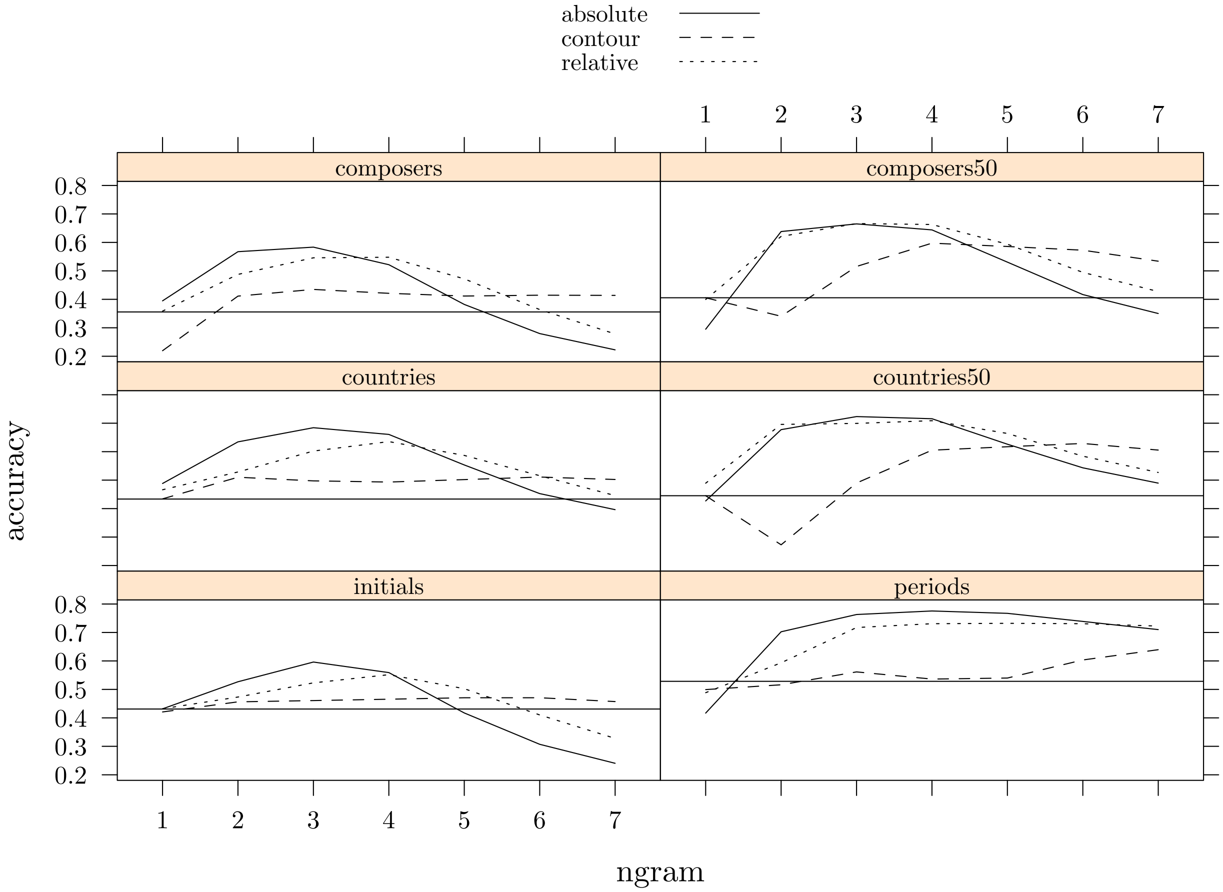


Figure 1. Accuracy of patterns of n-grams of one length. The horizontal line is the baseline.

the lowest. On inspecting the datasets, it turns out that there are strong relations between the different classification tasks. When cross-tabulating the *initials* task with the *countries* task, we can see that if for each of the countries classes (14 classes) we pick the most frequently co-occurring initials class (out of 17 classes), 71.8% of the data is described. Also, cross-tabulating the periods task (6 classes) and again taking the initials class (out of 17 classes) with the highest co-occurrence count, 55.3% of the pieces are explained.

Taking this into the extreme, we can also examine the composers task. Assume that we have a perfect classifier for this task. We can then learn a direct mapping from the class of composers to that of the initials. Obviously, this is possible, since there exists a simple relationship between the composer classes and the initials classes. In other words, if we know the composer, we can select the correct initials class perfectly.

However, during the initials task, the system does not have complete knowledge on the classes of the composer task. If we reverse the co-occurrence in the cross-tabulation, we see that when classifying into composers (50 classes) from the initials task (17 classes) it is still possible to classify 81.1% correctly. Similarly, we can correctly classify into the countries dataset (14 classes) 89.6% and into the period dataset (6 classes) 83.5% of the data. Overall, there is a large overlap between the classes from the different classification tasks.

Finally, we investigate the accuracy results when we allow patterns to have different lengths (specified within a range) in the classifier. The results of these experiments can be found in Figure 2.

Comparing the results (to fixed length n-grams in Figure 1), the curves are generally flatter and the results somewhat lower. We still see the expected increase in performance with a frequency threshold/reduction in classes, but no more peaks are evident. Again, the results for relative and absolute encodings are comparable, whereas contour performs significantly worse.

The reason for these flatter curves is that, as we can see in Figure 1, the longer n-grams occur less frequently (hence the lower accuracy), but in that case the shorter

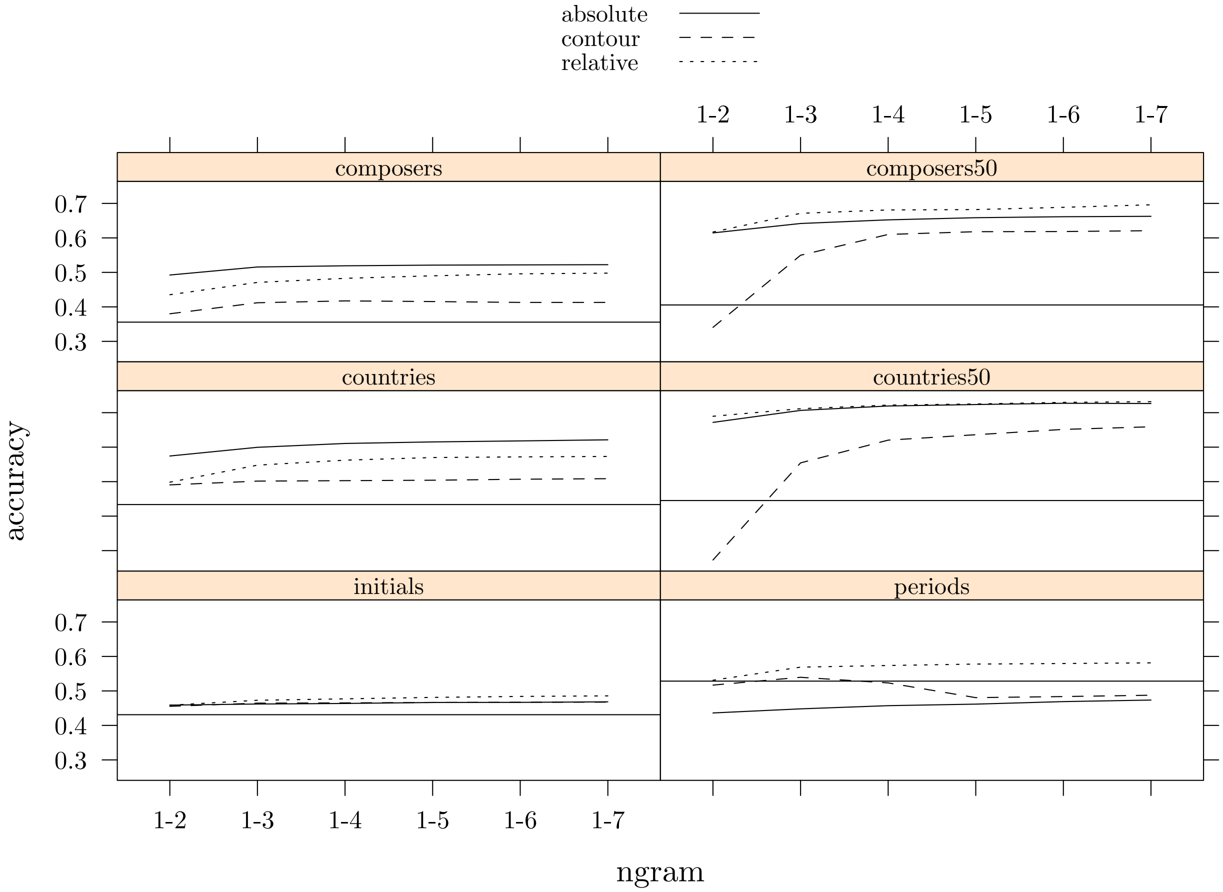


Figure 2. Accuracy of patterns of n-grams of a range of lengths. The horizontal line is the baseline.

n-grams take over. Furthermore, smaller n-grams may actually be contained in the longer n-grams, which then means that they are effectively used twice.

The encodings lead to similar results as in the single n-gram case: the difference between absolute and relative encodings are not statistically significantly different, whereas the contour encoding generally performs worse than the other encodings.

The range results also show some aspects that are harder to explain. Consider, for instance, the periods task where the absolute encoding (and the contour encoding with larger n-grams) falls below the baseline. We currently do not have an explanation for this.

For the task of classifying according to the first letter of the last name of a composer (the *initials* tasks), we now see the expected outcome: a flat curve (just above the baseline) which indicates that the patterns have only minimal influence on classification.

Comparing the use of fixed length n-grams and range-based n-grams, the results show that using n-gram ranges leads to more robust results. Taking the datasets and encodings into account, there is a statistically significant difference between the single or range n-grams.

5. Conclusions and future work

In this paper we have described a classification system that aims at finding regularly recurring patterns that have a high degree of discrimination among different classes. The well-known *tf*idf* metric is used to find and score the patterns. This score is also used to identify the best matching class for new, unseen sequences the system tries to classify.

To illustrate different aspects of the system, we have applied it to one dataset, but in different classification tasks. In particular, taking a dataset containing musical pieces, we provided classification tasks based on names of composers, the countries the composers come from, the musical periods the composers belong to and the initial letter of the last name of the composer.

We also described three different encodings of the music, namely absolute (absolute values of pitch and duration), relative (taking a previous note into account) and contour (direction of pitch and duration).

Overall, the system performs significantly better than the majority baseline. There is no difference between the absolute and relative encodings and both outperform the contour information (which encodes less information and hence has a much smaller vocabulary).

When allowing the system to use patterns of one fixed length only, the results are better, but the choice of the pattern length is important. Patterns that are too short or too long reduce the accuracy of the system. This is no problem when a range of lengths of patterns is allowed as the results are more robust. However, the overall results are lower.

The use of multiple classification tasks on the same dataset has nice properties, as it keeps the amount of data in each classification task constant. This allows us to experiment with different numbers of classes (per classification task). However, it also gives rise to problems. For instance, there is a direct relationship between the different tasks, which makes it difficult to identify exactly what the system is learning. This is illustrated in the initials dataset, which we intended as a sanity check, but instead performed reasonably well, as it (partly) coincided with other classification tasks.

In future work, we would like to experiment on a range of datasets, perhaps also outside the musical domain. The aim of using different datasets is to get a better grip on when exactly the system breaks down, in particular with respect to the size of the vocabulary.

Furthermore, we plan to explore different scoring metrics. At the moment, we have used one *tf*idf* metric, but alternatives exist (Manning et al., 2008, p. 118). This is particularly interesting because we use the *idf* component in the scoring metric in a very small document collection (since each class is considered a document). The choice for this component may have a large impact on the overall performance of the system.

References

Backer, E., & van Kranenburg, P. (2005). On musical stylometry: a pattern recognition approach. *Pattern Recognition Letters*, *26*, 299–309.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval.* Reading:MA, USA: Addison-Wesley Publishing Company.

Basili, R., Serafini, A., & Stellato, A. (2004). Classification of musical genre: a machine learning approach. *Proceedings of the fifth International Conference on Music Information Retrieval (ISMIR); Barcelona, Spain.*

Downie, J. S. (2003). Music information retrieval. *Annual Review of Information Science and Technology*, *37*, 295–340.

Garcia, P., & Vidal, E. (1990). Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 920–925.

Geertzen, J., & van Zaanen, M. (2008). Composer classification using grammatical inference. *Proceedings of the MML 2008 International Workshop on Machine Learning and Music held in conjunction with ICML/COLT/UAI 2008, Helsinki, Finland* (pp. 17–18).

Heaps, H. S. (1978). *Information retrieval: Theoretical and computational aspects.* New York, United States: Academic press.

Huron, D. (1997). Humdrum and kern: selective feature encoding. In E. Selfridge-Field (Ed.), *Beyond MIDI: The handbook of musical codes*, 375–401. Cambridge:MA, USA and London, UK: Massachusetts Institute of Technology Press.

Kyradis, I. (2006). Symbolic music genre classification based on note pitch and duration. *Proceedings of the 10th East European Conference on Advances in Databases and Information Systems; Thessaloniki, Greece* (pp. 329–338).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to infomation retrieval.* New York, NY, USA: Cambridge University Press.

Parsons, D. (1975). *The directory of tunes and musical themes.* Spencer-Brown.

Sapp, C. S. (2005). Online database of scores in the humdrum file format. *Proceedings of the sixth International Conference on Music Information Retrieval (ISMIR); London, United Kingdom* (pp. 664–665).

van Rijsbergen, C. J. (1979). *Information retrieval.* Glasgow, UK: University of Glasgow. 2nd edition, Printout.

van Zaanen, M., & Gaustad, T. (2010). Grammatical inference as class discrimination. *Grammatical Inference: Theoretical Results and Applications; Valencia, Spain* (pp. 245–257). Berlin Heidelberg, Germany: Springer-Verlag.

Exploratory Recommendations Using Wikipedia's Linking Structure

Adrian M. Kentsch, Walter A. Kosters, Peter van der Putten and Frank W. Takes
{AKENTSCH,KOSTERS,PUTTEN,FTAKES}@LIACS.NL
Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands

Keywords: graph theory, information retrieval, recommender systems, semantic relatedness, ranking algorithms

Abstract

This ongoing research addresses the use of page ranking for computing relatedness coefficients between pairs of nodes in a directed graph, based on their edge structure. A novel, hybrid algorithm is proposed for a complete assessment of nodes and their connecting edges, which is then applied to a practical application, namely a recommender system for books, authors, and their respective movie adaptations. Through relatedness, a level of surprise can be added to the recommender. The recommendation is created by exploring and discovering items of interest beyond the scope of books and authors. These items are then used in an explanatory manner to support the resulting recommendation. The chosen knowledge base is Wikipedia, a suitable source for both computing relatedness coefficients and applying them to the specific recommender system for reading material.

1. Introduction

Recently marking ten years since its launch, Wikipedia became a highly valuable and nearly complete online encyclopedia, with over three and a half million articles kept up-to-date by its active community of contributors (Wikipedia, 2011b). Also, in recent years, it became widely acknowledged as a fair source for reliable information. Beyond its rich content, Wikipedia is furthermore a well-structured collection of webpages representing article-entries that are created and updated to fit certain rules. Its link-structure, namely how articles within Wikipedia link to each other, is the main focus of this research. The motivation for choosing Wikipedia is given by its high level of information and reliability, as well as the particular rules involving the utility of inter-article links (Wikipedia, 2011c). As a consequence to these rules, links can be regarded as article keywords. Although not all links necessarily represent keywords, all words regarded as keywords will also have corresponding links. We assume, supported by other research (Milne & Witten, 2008), that keywords should have a high relatedness score with the article they represent.

Unlike other recommender systems (Melville & Sindhwani, 2010), we propose using Wikipedia's link structure as means to recommend related items to users. This approach is driven by the advantage to have an accurate cross-domain relatedness relationship between items. In other words, items for which relatedness is computed and recommendation is made should not necessarily belong to the same category. Furthermore, with our approach there is no cold start (Schein et al., 2002), a problem frequently occurring with other recommender systems for new users and items. In order to illustrate such a recommender, we propose applying it to a specific domain, which can be regarded as a subset of Wikipedia, namely the subset of thriller/crime fiction authors. This subset includes all linked articles up to a second degree link—links of links—among which the thriller/crime novels and existing movie adaptations would be present too. Such a subset is quite large, as it covers most of the strongly-connected Wikipedia articles, which in turn, as (Dolan, 2011) shows, almost cover the entire Wikipedia.

To give an example for recommending fiction authors, Tom Hanks, the actor, could represent to some users a strong enough reason to relate Dan Brown to Stephen King: he played the leading role in movie adaptations of books written by both authors. The same result may be revealed by collaborative filtering methods (Melville & Sindhwani, 2010), but those are not capable to explain to the user why such recommendation

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).

takes place, other than claiming it fits similar user behavior. Furthermore, although certain users may fit the same category of preference, they might still have their preference driven by distinct factors. A more personal motivation given to users when deriving a recommendation is therefore valuable.

The choice to compare items with each other through their respective Wikipedia articles is not driven only by finding motivation, but also by the ability to control the *surprise level* of the recommender. We argue that by selecting items scoring lower in relatedness than others, we can induce a "surprise" element, meaning that the user is familiar with such items less than with highly related ones, also making the derived recommendation less common (Onuma et al., 2009). This is what makes relatedness essential to the algorithm: it facilitates the ability to modify the surprise level. A thorough description of the recommender system will be offered in Section 4.

Semantic analysis and a text-based approach may be, in some cases, more suitable for computing relatedness between articles (Gabrilovich & Markovitch, 2007), but it certainly involves a higher level of complexity which, as research has shown (Milne & Witten, 2008), is not performing better in experiments than the simpler and more efficient link-based alternative. There are, however, several drawbacks to currently existing link-based algorithms for computing relatedness between articles, due to lack of completeness, which will also be addressed later.

In this paper, Wikipedia's set of articles and links is formally treated as a directed graph of nodes–articles, and edges–inter-article links. We treat Wikipedia both as a set of webpages and hyperlinks, and a set of documents and terms, often used as such in information retrieval (Manning et al., 2008), depending on what formulas are described.

Our main interest presented with this paper is to derive a desirable method for computing relatedness between articles that we can apply to the surprise level when offering recommendations. For this particular purpose, taking into account the entire linking-structure of articles is important. Similar research for link-based relatedness have been conducted, either directly on Wikipedia (Milne & Witten, 2008), or generally on graphs (Lin et al., 2007). These either only partially take into account linking properties (Milne & Witten, 2008), and are therefore incomplete, or depend on a hierarchical structure of articles and on walking through the entire graph to compute relatedness (Lin et al., 2007), thus being conditional and expensive. We aim for a complete, unconditional and inexpensive algorithm that needs no further information than links of links for computing relatedness.

We use the following notations throughout the paper: For the article-node A, L_A represents the set of all articles to which A links, while B_A represents the set of all articles that link to A. These are also known as *outlinks* and *inlinks* respectively (Langville & Meyer, 2006). However, in Wikipedia the latter is called the set of *backlinks* of A (Wikipedia, 2011a), whereas the former is called the set of *links*. For this paper, we opt for Wikipedia's vocabulary. We also represent the link-edge oriented from A_1 to A_2 by $A_1 \rightarrow A_2$ and the set of all Wikipedia articles by W. Furthermore, all formulas that we present here are adapted by us to fit the above notations.

The remainder of this paper is divided into five sections: Section 2 describes the link-structure of Wikipedia's corresponding graph and how relatedness between articles is computed; Section 3 takes a step further towards ranking nodes in a graph and weighting certain relatedness computations more than other; Section 4 presents the practical application of Wikipedia's graph-relatedness to the actual recommender system for thriller/crime fiction; Section 5 illustrates the advantages of such an algorithm through concrete examples; and finally Section 6 presents a conclusion and proposed future work.

2. Link Analysis and Relatedness

There are several scenarios that carry information regarding the way two articles, A_1 and A_2, might relate to each other, as illustrated in Figure 1. One scenario is represented by the *direct link* between the two articles. If one direct link exists, formally when either $A_2 \in L_{A_1}$ or $A_1 \in L_{A_2}$, it intuitively implies the articles are more related than if it does not exist. Another scenario, implying an even stronger relatedness, would be the *reciprocal link*, meaning that both articles directly link to each other. The *directed paths* would represent a third scenario, meaning there is no direct link, but there are one or more directed paths with intermediate nodes. In practice, given that Wikipedia represents a graph of strongly connected articles, directed paths exist for almost every pair of articles, the average shortest path being of length 4, as found by (Dolan, 2011). Thus, we believe that only the directed paths with one node in between significantly contributes to relatedness. This in other words is equivalent to a shortest path, unweighted, of length 2, occurring when $L_{A_1} \cap B_{A_2} \neq \emptyset$ or $L_{A_2} \cap B_{A_1} \neq \emptyset$.

Two more scenarios regarding an intermediate node are represented by *shared links* and *shared backlinks*. They formally occur in the following situations: in the former case when $L_{A_1} \cap L_{A_2} \neq \emptyset$, and in the latter case when $B_{A_1} \cap B_{A_2} \neq \emptyset$. In information retrieval these are also known as *co-reference* and *co-citation* respectively (Langville & Meyer, 2006). Usually, the more shared links or shared backlinks between two articles, the more related the articles are. There are several methods to normalize a relatedness coefficient based on the number of shared articles, the *Jaccard index*, computed by dividing the size of the intersection by the size of the union, being one of them (Lin et al., 2007); but a more complex approach that also relates the result to the size of the entire set of articles is preferred. (Milne & Witten, 2008) proposes two different methods for computing relatedness, one for shared links and one for shared backlinks, both methods being commonly used in information retrieval with text and queries. For shared backlinks, the *Normalized Google Distance* $NGD\left(A_1, A_2\right)$ between articles A_1 and A_2 is defined by (Milne & Witten, 2008) as:

$$NGD\left(A_1, A_2\right) = \frac{\log\left(\max\left(\left|B_{A_1}\right|, \left|B_{A_2}\right|\right)\right) - \log\left(\left|B_{A_1} \cap B_{A_2}\right|\right)}{\log(|W|) - \log\left(\min\left(\left|B_{A_1}\right|, \left|B_{A_2}\right|\right)\right)} \tag{1}$$

where W stands for the set of all Wikipedia articles. Note that the size of each set is taken into account, representing the number of backlinks. A shared backlink by definition implies there is an article containing both terms. The Normalized Google Distance calculates how related terms are by their occurrence separately and together in all other webpages (Cilibrasi & Vitanyi, 2007), which we agree is suitable for our case. The values for this function range from 0 to ∞, so the result needs to be further normalized to range as a coefficient, from 0 to 1, not as distance. (Milne & Witten, 2008) proposes to invert the values between 0 and 1 and ignore all values that fall beyond, which, given that a distance higher than 1 implies *negative correlation* (Cilibrasi & Vitanyi, 2007), is a fair assumption. We define the *relatedness coefficient between A_1 and A_2 through backlinks $RC_B\left(A_1, A_2\right)$* as:

$$RC_B\left(A_1, A_2\right) = \begin{cases} 1 - NGD(A_1, A_2) & 0 \leq NGD(A_1, A_2) \leq 1 \\ 0 & NGD(A_1, A_2) > 1 \end{cases} \tag{2}$$

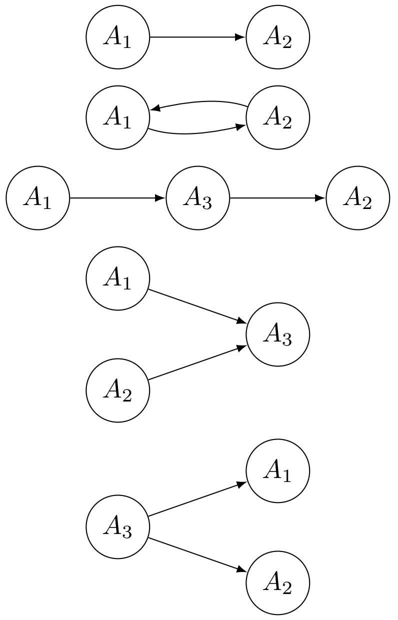


Figure 1. The *five scenarios* for A_1—A_2 relatedness, top–down: the *direct link*, the *reciprocal link*, the *directed path*, the *shared link* and the *shared backlink*.

For relatedness via shared links, (Milne & Witten, 2008) proposes the *cosine similarity of* `tf` × `idf` *weights*, another method popular with information retrieval (Manning et al., 2008) measuring how important terms are to documents containing them, in our case links to articles. Basically, each shared link is first weighted using the `tf` × `idf` formula as follows:

$$w_{A' \to A} = \left(\texttt{tf} \times \texttt{idf}\right)_{A' \to A} = \frac{1}{\left|L_{A'}\right|} \times \log\frac{|W|}{\left|B_A\right|} \tag{3}$$

where A is the *term*–article or link, and w_A is its *weight*. The formula is simplified twice, firstly because the `tf` formula does not need to count the number of times the same link occurs in an article because in Wikipedia this should only happen once if it exists. (Milne & Witten, 2008) further simplifies this formula, completely eliminating `tf` from the equation, after observing it does not affect the final result. Thus, the value of the weight becomes the `idf` of the term–article, independent from the *document*–article:

$$w_A = \left(\texttt{tf} \times \texttt{idf}\right)_A = \texttt{idf}_A = \log\frac{|W|}{\left|B_A\right|} \tag{4}$$

All links are combined and consequently normalized with the following expression form for the *cosine similarity*, representing the *relatedness coefficient between A_1 and A_2 through links $RC_L\left(A_1, A_2\right)$*:

$$RC_L\left(A_1, A_2\right) = \frac{\sum_{A \in L_{A_1} \cap L_{A_2}} {w_A}^2}{\sqrt{\sum_{A \in L_{A_1}} {w_A}^2 \times} \sqrt{\sum_{A \in L_{A_2}} {w_A}^2}} \tag{5}$$

We chose to also simplify this equation from its standard form, thus illustrating the summations of weights for existing links only, the weights for non-existing links being 0 and therefore not included above.

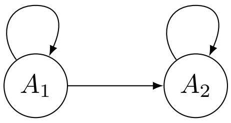


Figure 2. By assuming A_1 and A_2 both link to themselves, a direct link $A_1 \rightarrow A_2$ implies that A_1 becomes a *shared backlink* and A_2 becomes a *shared link* between A_1 and A_2.

(Milne & Witten, 2008) shows that both methods to compute relatedness between articles on Wikipedia perform better in experiments than semantic methods and proposes an arithmetic average of the two relatedness coefficients to be computed for the overall coefficient. We believe this is not always justified, and thus introduce a weighted average $\overline{RC}_{(L,B)}$ instead:

$$\overline{RC}_{(L,B)} = \alpha\, RC_L + (1{-}\alpha)\, RC_B \tag{6}$$

where α is a variable between 0 and 1, depending on which relatedness should be weighted more than the other. Furthermore, (Milne & Witten, 2008) omits the directed or reciprocal link scenarios, which should also be treated. In fact, if such scenarios occur, their formulas decrease the relatedness coefficient instead of increasing it. Our first "fix" is to assume all articles link to themselves too, as illustrated in Figure 2. We keep this assumtion for the remainder of this paper: $A \in L_A$ and $A \in B_A$. A direct link, however, should have a stronger influence on relatedness than just an extra shared link. Therefore, we introduce another variable, β, and define the overall $\overline{RC}$ as:

$$\overline{RC} = \overline{RC}_{(L,B)} + \beta\left(1 - \overline{RC}_{(L,B)}\right) \tag{7}$$

In the next section, we present the derived formulas for variables α and β. The directed paths will also be treated and included in β's formula.

3. Page Ranking and Weighting

When computing relatedness through shared links and shared backlinks, information about each article's "parents", "children" and "other parents of its children" is required, which means backlinks, links and backlinks of links. This is formally known as the article's *Markov blanket.* In this section we will also require the "other children of parents", the links of backlinks, for computing what is known as a particular type of a *Markov chain* used for ranking webpages, namely the *Google PageRank.*

Algorithms that iterate throughout the whole graph are used for webpage ranking, but also for computing relatedness. Though they are expensive, they can discover useful information. Google's PageRank for example computes the probability that a random surfer visits a certain article (Langville & Meyer, 2006), thus determining which pages attract more visitors. We are using a simplified PageRank, iterating only once and only through the subgraph defined by the article itself, its backlinks and the links of its backlinks. The article's links are also included, because of our assumption that all articles link to themselves too, making the computational space an inverse of a Markov blanket. The PageRank PR_A for article A is formally given as:

$$PR_A = \sum_{A' \in B_A} \frac{1}{|L_{A'}|} \tag{8}$$

This measure indicates the chance that someone visiting a backlink of an article eventually arrives on that particular article too. We believe this is a good candidate for variable α from Equation 6 not only because it can determine whether backlinks are more important than links for relatedness, but also because it adds what Equation 2 was ignoring: weighing those backlinks similarly to how Equation 4 weighs links before combining them in Equation 5. With another popular ranking algorithm called *HITS*, this is equivalent to determining whether articles have a higher *authority* score or a higher *hub* score (Kleinberg, 1999).

A further application for PageRank is found for the directed paths, whose influence on relatedness is determined by how much, say, A_1 contributes to the rank of A_2 via the respective paths. The more paths there are, the higher the relatedness will be. Keeping consistency with our chosen computational space, these paths will be restricted to a maximum length of 2, having only one intermediate node, longer paths being ignored. The PageRank can therefore not only determine α, but also β, which can then be used for both the *direct link* and the *directed paths.* As Equation 8 shows, the rank transferred from one page to another via its links is equally distributed, becoming the rank of the page divided by its number of links. With this observation in mind, we can define the PageRank $PR_{A_1 \rightarrow A_2}$ of a path $A_1 \rightarrow A_2$ as follows:

$$PR_{A_1 \rightarrow A_2} = \tfrac{1}{|L_{A_1}|}\left(|\{A_2\} \cap L_{A_1}| + \textstyle\sum_{A \in L_{A_1} \cap B_{A_2}} \tfrac{1}{|L_A|}\right) \tag{9}$$

Earlier we mentioned that similar iterative algorithms exist to directly compute relatedness rather than ranking, *SimRank* (Jeh & Widom, 2002) and *PageSim* (Lin et al., 2007) being two of them. These can be considered related research, but unlike our approach, they are even more expensive than PageRank, computing values for pairs of nodes instead of just nodes. Furthermore, they also require a hierarchically structured graph (Jeh & Widom, 2002), incompatible to Wikipedia's. However, the improvement to SimRank used by (Lin et al., 2007) with the introduction of PageSim is similar to our own improvement to (Milne & Witten, 2008). Concretely, (Lin et al., 2007) also implement a version of the path PageRank and take advantage of the complete linking structure. Other than this, their research has a different scope.

In order to determine α from Equation 6, we need to take into account the PageRanks for A_1 and A_2 representing the articles compared. We do this by taking the arithmetic average of the two articles, but we also need to normalize the result. In theory, using our formula, PageRank can take values ranging from 0 to the number of all articles involved. Since there is only one iteration, the articles involved represent the union of backlinks B_{A_1} with backlinks B_{A_2}. Furthermore, before iteration, all PageRanks are equal to 1. After normalization, these values should become equivalent to a minimum of 0, a maximum of 1 and an initial or neutral value of 0.5. Therefore, the following equations are considered to determine α:

$$\alpha' = \frac{1}{2}\left(PR_{A_1} + PR_{A_2}\right) \qquad (10)$$

$$\alpha = \frac{\alpha'\left(|B_{A_1} \cup B_{A_2}| - 1\right)}{\alpha'\left(|B_{A_1} \cup B_{A_2}| - 2\right) + |B_{A_1} \cup B_{A_2}|} \qquad (11)$$

where α' is the value before normalization. The PageRank of a path does not require normalization, because it has no neutral value and takes values ranging between 0 and 0.5. Thus, the proposed formula for β, is simply the arithmetic average of the path PageRanks:

$$\beta = \frac{1}{2}\left(PR_{A_1 \to A_2} + PR_{A_2 \to A_1}\right) \qquad (12)$$

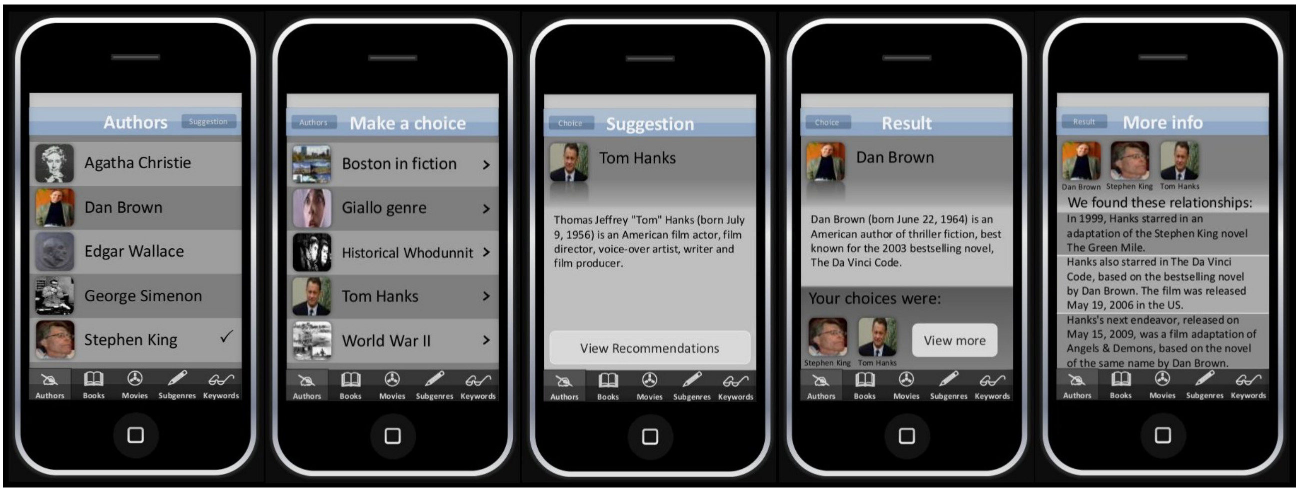
To sum up, given two articles A_1 and A_2, their relatedness coefficient via backlinks is derived in Equation 2, and via links in Equation 5. Then, Equation 6 offers a weighted average formula between the two types of relatedness, whose α is computed using the normalized PageRank. Finally, Equation 7 takes into account the entire linking structure of articles, by considering direct links and directed paths too. This is done through β, which uses the path PageRank. When α is equal to 0.5, it equally weighs relatedness over links with the one over backlinks, and when β is 0, it means there are no direct links and no directed paths.

4. Recommender System

As mentioned in the beginning, the Wikipedia-based graph of articles and their corresponding computed relatedness coefficients are applied to an interactive and exploratory recommender system. This represents a learning mechanism for both the user and the recommender, through which it is discovered what exact items of interest drive the user to the end-resulting recommendation. The domain of thriller/crime fiction authors is chosen to exemplify this recommender as follows: the user is first prompted with a list of authors from which to select the ones that he or she is interested in; next, a list of related items that do not belong to the category of thriller/crime authors is shown, from which the user is again invited to select what is of interest; the iteration can be repeated if still necessary—if more authors could still be suggested—or the user could opt for an immediate recommendation; when this recommendation is shown, it will be complemented by an overview of all selected items and how they relate to each other, to the initially selected thriller/crime authors, and to the resulted recommendation. This is done by simply quoting the paragraphs from which the mentioned articles link to each other.

Because Wikipedia's policy is to only link once from an article to another, searching for paragraphs containing the reference actually implies searching for keywords, not links. Sometimes these keywords are more difficult to find because alternative text can be used for links, also known as anchor text or link label. For example in "Dan Brown is American" the word "American" is an *alternative text* linking to the "United States" article. There are several methods to improve this search, the simplest being to only look for keywords representing the linked article title—"United States"—and its alternative text—"American"—neglecting the rest. A semantic approach is sometimes more suitable (Gabrilovich & Markovitch, 2007), but it falls beyond the scope of our research.

A special feature of our recommender system is the *surprise level*, which can be easily modified by choosing between different ranges of relatedness values. We claim that if the highest related articles are chosen, say the ones with a coefficient above 0.750, the end result will correspond to the prediction of other recommender systems. However, if we limit relatedness values to a range between 0.250 and 0.500, more *surprising* (Onuma et al., 2009) intermediary items will be shown, but the end result can still be justified using the same approach that mentions how all articles relate to each other. In fact, besides filtering out articles that are nearly unrelated—with a coefficient below 0.250—the main use of the relatedness coefficient is to be able to modify this surprise level. If the user is interested in finding the most similar author to his preference, thus aiming for low surprise, then he or she will receive a



Thomas Jeffrey "Tom" Hanks (born July 9, 1956) is an American film actor, film director, voice-over artist, writer and film producer.

Dan Brown (born June 22, 1964) is an American author of thriller fiction, best known for the 2003 bestselling novel, The Da Vinci Code.

In 1999, Hanks starred in an adaptation of the Stephen King novel The Green Mile.

Hanks also starred in The Da Vinci Code, based on the bestselling novel by Dan Brown. The film was released May 19, 2006 in the US.

Hanks's next endeavor, released on May 15, 2009, was a film adaptation of Angels & Demons, based on the novel of the same name by Dan Brown.

Figure 3. Mobile Application screenshots exemplifying the recommendation path from author "Stephen King" to author "Dan Brown" when selecting "Tom Hanks" as keyword.

list with the most related items. If, however, the user aims for a more surprising recommendation, and consequently a more exploratory experience, the surprise level is increased, such that the most commonly known items will be hidden in favor of the least expected ones.

Our approach also presents several other advantages when compared to popular recommender systems such as collaborative filtering. The problem of *cold start* (Schein et al., 2002), for example, which occurs when the system does not hold sufficient information on users and items to reach a recommendation, is not present with our approach because we extract all necessary information from Wikipedia. Also, our algorithm works when no input is selected, when, in the presented case, no fiction author is initially selected by the user. This for example may happen when the user does not yet know any author to be able to select from. Since it is already known what articles highly relate to authors, or what articles are shared among two or more authors, a selected list of articles can always be offered given the user's preference for the surprise level: more expected or more surprising.

All relatedness coefficients between authors and their "neighbors"—links and backlinks—are precomputed and stored in a database as soon as the required information is synchronized with Wikipedia's, guaranteeing a quick response for the recommender system. We call these "neighbors" *keyword-articles* and we precompute their relatedness not only to authors but to pairs of authors too, therefore knowing immediately which of them relate most to each other. We can also precompute related keyword-articles to more than two authors taken together, although for our recommender and for simplicity we prefer to keep them paired: one author from the list chosen by the user, and the other from the list of potential results. After author selection, we display the keyword-articles in the order of relatedness with pairs of authors.

Relatedness with a pair should not only take into account the relatedness computed with each author separately, but also how close to each other these coefficients are. Thus, we define RF to be the *relevance factor* and RC the *relatedness coefficient between term A and the pair* $\{A_1, A_2\}$ as follows:

$$RF\left(A, \{A_1, A_2\}\right) = \frac{\min(RC(A,A_1),RC(A,A_2))}{\max(RC(A,A_1),RC(A,A_2))} \qquad (13)$$

$$RC(A,\{A_1,A_2\})=RF(A,\{A_1,A_2\})\times\frac{RC(A,A_1)+RC(A,A_2)}{2} \qquad (14)$$

Therefore, the higher and closer to each other the two relatedness coefficients are, the higher the combined relatedness is.

Our proposed algorithm for the recommender system, which we are implementing for a website and a mobile application, is as follows: first let the user select one or more fiction authors from the input and choose whether the recommendation should be expected or surprising; then for all selected authors, take all keyword-articles shared with unselected authors, order them by the relatedness coefficient with the respective pair of authors and display the ones fitting the surprise criteria for a new selection; let the user select one keyword-article from the list and reveal the unselected author linked to it as recommendation; if more results fit the surprise criteria, opt for the highest relatedness; for the result, display all articles that have been involved together with all paragraphs in which the ar-

Table 1. Relatedness of "Dan Brown", "Stephen King", and both authors taken as pair, with their shared links and backlinks, regarded as keyword–articles.

Article Title	Dan Brown	Stephen King	both as pair
James Patterson	0.625	0.536	0.497
Tom Hanks	0.501	0.475	0.462
Ron Howard	0.624	0.420	0.351
The Da Vinci Code	0.978	0.427	0.306
Akiva Goldsman	0.558	0.368	0.305
Arthur Machen	0.343	0.561	0.276
Left Behind	0.586	0.339	0.267
Roger Ebert	0.224	0.398	0.175
New England	0.167	0.252	0.138

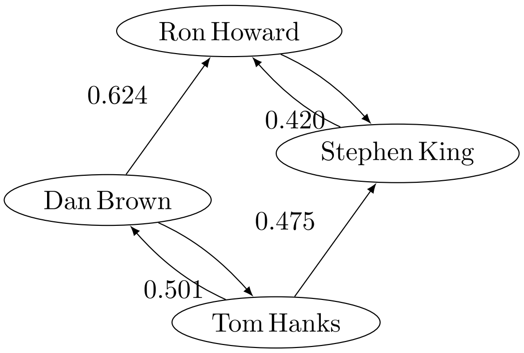


Figure 4. The corresponding articles for actor "Tom Hanks" and director "Ron Howard" are shared by fiction authors "Dan Brown" and "Stephen King".

ticles linked to each other. A few screenshots from our planned mobile application are shown in Figure 3.

5. Illustrative Example

In this section, we provide an illustrative example, elaborating on the images from Figure 3. We take the corresponding articles for authors "Dan Brown" and "Stephen King" together with their keyword-articles. The two authors have many shared links and backlinks, namely 11 shared links and 27 shared backlinks. Compared to the high number of links and backlinks that each has, 122 links + 344 backlinks for Dan Brown and 442 links + 2044 backlinks for Stephen King, their shared ones are few, but sufficient and significant. Table 1 lists some of them together with their relatedness scores, computed using Equation 7 for article-to-article relatedness and Equation 14 for article-to-pair. This situation applies to most pairs of authors, preventing the recommender system from displaying many results.

From the lists of shared links and backlinks we selected two representative keyword–articles that also happen to be among the highest in relatedness with both authors, as shown in Figure 4. Their relatedness coefficients are computed using Equation 7, which we designed to take into account the entire linking structure of compared articles. Note that these coefficients do not belong to link-edges, which can be reciprocal or even missing, but to the compared articles. In Figure 4, we just added these coefficients between the articles for which we computed relatedness.

There are of course a few articles that relate significantly more to one author than to the other. For example, "The Da Vinci Code", novel written by Dan Brown, relates 0.978 to Dan Brown, as expected, and only 0.427 to Stephen King. It is arguable whether "The Da Vinci Code" should be given more importance, say, than "Tom Hanks". Therefore, applying Equation 14 to compute how related "The Da Vinci Code" and "Tom Hanks" each are to the pair of articles ("Dan Brown", "Stephen King"), we obtain for "The Da Vinci Code" 0.306 and for "Tom Hanks" 0.462, concluding that "Tom Hanks" relates more to the two authors taken as pair than "The Da Vinci Code". In Table 1 the relatedness with the pair, computed using Equation 14, is shown in the last column.

In the example from Figure 4, the actor Tom Hanks and the director Ron Howard score quite well for the pair Dan Brown and Stephen King. This means that if a user likes Stephen King and then also likes Tom Hanks, he or she receives Dan Brown as recommendation with the reasoning that Tom Hanks played the leading role in Dan Brown's "The Da Vinci Code" movie trilogy, and also played the leading role in Stephen King's movie adaptation "The Green Mile". Similarly, Ron Howard directed the movies from "The Da Vinci Code" series, and is currently directing a TV series, "The Dark Tower", written by Stephen King. Their relatedness scores are quite similar, though it is interesting to observe, despite their similarity, that one is a shared backlink, while the other is a shared link. This illustrates that Wikipedia does not follow a hierarchical structure, links and backlinks being equally valuable as keywords.

Additionally, by looking solely at this graph, it can be observed that the PageRank of "Ron Howard" is equal to the one of 'Stephen King' and higher than the PageRank of the other two articles. To be more precise, "Ron Howard" and "Stephen King" would both have a PageRank of 1.333, whereas the PageRanks for "Dan Brown" and "Tom Hanks" would be 0.667 each. In this case, the *weighted average* from Equation 6 required to compute relatedness between "Dan Brown" and "Stephen King" corresponds to an arithmetic average, α being equal to 0.5. After also applying Equation 7, this relatedness has a coefficient of 0.503.

6. Conclusion and Future Work

We have shown throughout this paper a useful method to compute relatedness between nodes in a graph and to implement it in a recommender system. We used Wikipedia as the knowledge base and we exemplified our recommendations on thriller/crime fiction authors. We demonstrated that our approach has significant advantages over more classical approaches such as collaborative filtering. We also adapted and improved the relatedness measurements from related research, taking full advantage of the linking structure and at the same time keeping computation inexpensive. Finally, we discussed the *surprise level*, a feature of the recommender that allows the user to choose how surprising the results should be; and we also presented the *relevance factor*, allowing a better assessment of shared keyword-articles between selected and resulting authors.

As this is an ongoing research, future work involves further evaluation methods. We are currently assessing our relatedness algorithm and its impact on the performance of the recommender system and the surprise level, by comparing it with relatedness based on fewer linking properties, such as shared links and backlinks. We also plan to evaluate our recommender system against leading algorithms used on very large user-bases, in order to measure how similar the results are. Furthermore, we intend to test the recommender on both expert users and unknowledgeable users, in order to assess their satisfaction with our approach. Finally, we work towards launching the website and mobile application intended for public use.

Acknowledgments

The fourth author is supported by the NWO COMPASS project (grant #612.065.92).

References

Cilibrasi, R. L., & Vitanyi, P. M. B. (2007). The Google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, *19*, 370–383.

Dolan, S. (2011). Six degrees of Wikipedia. `http://www.netsoc.tcd.ie/~mu/wiki`.

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *Proceedings of the 20th international joint conference on Artifical intelligence* (pp. 1606–1611).

Jeh, G., & Widom, J. (2002). Simrank: A measure of structural-context similarity. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 538–543).

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal ACM*, *46*, 604–632.

Langville, A. N., & Meyer, C. D. (2006). *Google's PageRank and beyond: The science of search engine rankings.* Princeton University Press.

Lin, Z., Lyu, M. R., & King, I. (2007). Extending link-based algorithms for similar web pages with neighborhood structure. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 263–266).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval.* Cambridge University Press.

Melville, P., & Sindhwani, V. (2010). Recommender systems. In *Encyclopedia of machine learning*, chapter 705, 829–838. Boston, MA.

Milne, D., & Witten, I. H. (2008). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. *Proceedings of the First AAAI Workshop on Wikipedia and Artificial Intelligence* (pp. 25–30).

Onuma, K., Tong, H., & Faloutsos, C. (2009). Tangent: A novel, 'Surprise me', recommendation algorithm. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 657–666).

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Eetrieval* (pp. 253–260).

Wikipedia (2011a). Help: What links here. `http://en.wikipedia.org/wiki/Help:What_links_here`.

Wikipedia (2011b). Wikipedia: About. `http://en.wikipedia.org/wiki/Wikipedia:About`.

Wikipedia (2011c). Wikipedia: Manual of style (linking). `http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_(linking)`.

A Feature Construction and Classification Approach to Pixel Annotation

Arno Knobbe KNOBBE@LIACS.NL
LIACS, Leiden University, the Netherlands

Pieter Hoogestijn PIETER@HOOGESTIJN.NL
Everest, 's Hertogenbosch, the Netherlands

Durk Kingma DURK@ADVANZA.NL
Advanza, Utrecht, the Netherlands

Abstract

In this paper, we present a Machine Learning approach to an important problem in Computer Vision: the pixel annotation task. This task is concerned with assigning one of a small set of predefined classes to every pixel in an image. The classes can be any concept defined by the user, but in our case typically will represent particular surface-types. A logical example that we will be considering is the classification of road-surface. We will demonstrate how likely areas of tarmac can be identified using a trained classifier, and demonstrate how this low-level information may be used for navigation purposes. An important characteristic of the selected approach is the importance of feature construction. We define a large collection of features based on different aspects of a pixel and its environment. These include concepts such as colour, edges, textures and so on. Additionally, we will include a range of features derived from the spatial information available from stereo images. The information about surface location and orientation obtained through stereo vision may hold important clues about the pixel class that cannot be obtained from single images.

1. Introduction

Despite many recent advances in the field of Computer Vision, it is still very hard for computers to truly understand what is happening in a given image. Quite a number of steps are necessary before the detailed low-level information can be turned into high-level semantic descriptions of the scene at hand. Most methods start by applying filters to the original image, for example for edge detection. These local features can then be combined into more intermediate-level objects such as lines and corners. Further steps may finally lead to the recognition of objects and their spatial relationships, although this general problem is still only solved for very restricted settings. In this paper, we aid this complex process, by presenting a solution for one of the necessary steps, namely the recognition of specific surface-types. This is done by training a learning algorithm on a database of previously annotated images.

Clearly, being able to assign types of material or surface to certain areas of the image is a very useful step in this process. If a certain procedure is able to find with a reasonable level of reliability which parts of an image represent, say, fur, then the task of recognizing (the location of) dogs is already partly solved. Next to being a good starting point for object recognition and image understanding, the ability to find areas of a given type can also be used for lower-level reasoning about possible actions in the observed scene. One of the tasks we consider in this paper is that of recognizing road surface. Clearly, an efficient form of path planning can be obtained from the knowledge of where exactly the road is heading, without exactly understanding the three-dimensional details of the scene and the location of possible obstacles. Minor errors in the assignment of

Appearing in *Proceedings of the 20^{th} Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).

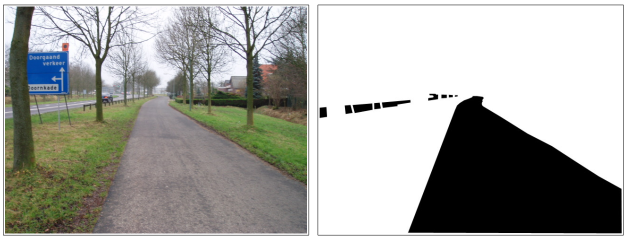


Figure 1. The original image and the binary annotation related to the concept 'road surface'.

road surface will be ignored because for path planning, only the overall identification of safe areas is relevant.

In this paper, we take a machine learning approach, by treating the surface identification problem as a classification problem. It is assumed that a (relatively small) number of images is annotated by hand, and a classifier is trained on the (large set of) examples obtained from these images. In the basic setting, the classification problem is assumed to be binary: a particular region either belongs to a certain class or it does not. Hence the annotation of a training image consists of a two-coloured image of the same proportions (see Figure 1). The binary setting can be easily upgraded to a multi-class problem by inducing multiple classifiers, one for each class, and then combining predictions.

The classification problem at hand is challenging for a number of reasons. First of all, it is unlikely that basic features such as RGB-values of individual pixels will be very discriminative. For example, the colour of tarmac is likely to be similar that of concrete, or even the sky on some days. Therefor, a considerable amount of feature construction will be necessary, in order to include local properties such as texture and gradient information. Section 3 lists the range of features that we consider. The problem is also challenging simply because of the complexity of the outside world. There may be many variations in the actual image, depending on weather and lighting conditions. Clearly, there is a need for robust features that are somewhat insensitive to such variations. For example, we require features that are not very sensitive to shadows on the road from, say, trees along the road. The *colour saturation* [1] (an inverse measure for the 'greyness') is such a lighting-invariant feature. Finally, the positive examples in the training data may form non-convex, or even noncontiguous regions in the input space, and classifiers need to account for such complexities. In the road surface task, for example, both the road markings and the actual tarmac need to be classified positively, even though these examples clearly lie in separate areas of the RGB space. In our second task, the identification of traffic signs, a range of different, bright colours may be encountered, each of which should be classified positively.

One special property of our approach (to the best of our knowledge unique) is the inclusion of spatial information in the list of features. In the simplest case, this involves the x and y-coordinate of the pixel, such that the classification procedure can estimate the likelihood of certain objects appearing in certain segments of the image (e.g. roads never appear above the horizon). More importantly, we are including 3D information obtained from stereo images. The Computer Vision package we are using, Harmonii [2], determines the three-dimensional location of each pixel in world-coordinates, using a disparity algorithm that determines the distance between corresponding pixels in the two images. This 3D information, and derived features such as the direction of the surface normal and the flatness of the surface, may help improve the predictive accuracy considerably. The down-side of including stereo information is the non-negligible computational cost of stereo vision. We will compare the results with and without 3D information, and will demonstrate that in some cases, the classifier-based approach may actually be an efficient replacement for the expensive stereo algorithm.

2. Problem Definition

The problem scope of this paper is the *classification of pixels* within a two-dimensional image, also referred to in literature as *pixel annotation* [3]. It is important to keep in mind that this is distinct from the related problem of image annotation [4], which is about classification of a whole image and automatic finding of the right image labels. The task of a pixel classifier, on the other hand, is to find the correct labelling of a pixel, given a finite set of possible classes. Since pixel classification is giving a separate class to each pixel, this is both intrinsically harder then image annotation, but also potentially more powerful since its result is more detailed in semantics.

In the field of Machine Learning, supervised learning is concerned with classification of unseen data given some training data with class labels. The training data is pairs of input vectors and class labels (this as opposed to unsupervised/semi-supervised learning, which is concerned with learning from (partially) unlabeled data). The job of the classifier is to generate a model using the available training data, and use this model to predict class labels for unseen, unlabeled data.

We are concerned with two separate classification tasks. The first task, *Tarmac*, is to distinguish road from non-road: a typical task in a robotic planning setting, where route planning is only acceptable over road surface. Note that the target concept not only includes actual tarmac, but also road markings, etc. Our second task, *Sign*, is to correctly pick out the pixels belonging to traffic signs. A challenge here is the many different colours that may appear on signs. Fig. 1 shows an example of the kind of signs we intend.

For each task, training data was generated by manually crafting monochrome overlays, black being the positive class (e.g. road), white being the negative class (e.g. non-road). This effectively created annotated training images. For both tasks, a moderate number of images were taken in an automotive setting with different types of tarmac and lighting conditions. A deterministic sliding window mechanism was used to generate the individual training samples, resulting in about 11k training points per image.

One important question is: what input vector should we use? An obvious answer is to simply extract the pixel values (RGB or HSV values) of the window surrounding the pixel. For example, we take a window of 5 by 5 pixels surrounding the pixel we want to classify, and use their RGB values as input vector, resulting in $5 \cdot 5 \cdot 3 = 75$ values as input vector and use these to learn and predict the class of the central pixel. In theory, if the window size is sufficiently large, the information provided should be enough. However, most classification algorithms in literature require features that are individually discriminative. In most applications, individual pixel values may not be very informative, so classification algorithms can fail to generate good models on such data.

To solve this problem, we add a pre-processing phase that constructs a large number of features. Each feature takes as input some information from a window surrounding the pixel, and outputs a scalar value calculated from this window. As these features are calculated as combination of pixel values, at least some features should have good discriminative power. Our complete list of features will be described in the next section.

An interesting research question is the value of depth (3D) information. The software we used, Harmonii [2], provides a robust stereo vision engine that generates a point cloud from two images, effectively providing 3D information for pixels in our images. Several features use this spatial information, and we will report the improvement in classification accuracy in the results section.

3. Feature Construction

As input for the classifier, a number of features were extracted from the images. Besides using the features already provided by the Harmonii software, we also used several combinations of these features.

In the subsections below, we explain all the features used in our experiments. The features are divided into a number of feature groups, which in our experiment could be toggled on and off for usage in building the classification model. In total, 137 features were defined to capture the properties of a particular pixel and its neighbourhood.

3.1. Colour features

The colour of the pixel and its neighbourhood is the most obvious information available to the classifier. The following 52 colour features were defined:

3.1.1. RGB and HSV, Grey.

The basic RGB-values were taken, as well as the derived HSV-values (*Hue*, *Saturation* and *Value* [1]). Additionally, we took *Grey*, which is the mean of the R, G and B components.

3.1.2. Colour intensities of red, green, blue.

Apart from the absolute colour levels, we also included the relative colour intensity of the colours red, green and blue:

$$R_i = \frac{R}{G+B}, \quad G_i = \frac{G}{R+B} \quad \text{and} \quad B_i = \frac{B}{R+G}$$

where R_i, G_i and B_i are the intensity feature values.

3.1.3. Mean μ and variance σ^2 of RGB, HSV, Grey.

A $N \times N$ window around the central pixel was taken, and for each of the seven RGB/HSV/Grey components, the mean and variance were calculated. Note that the mean is essentially analogous to the result of a homogenous convolutional blur operation, and is known to be effective against Gaussian noise since the deviations are normally distributed, and are relative to the original value.

3.1.4. Colour Histograms.

Histograms are used to count the quantity of pixel values from within a certain value range, for each of the seven elements (R, G, B, H, S, V and Grey). To keep the amount of features within acceptable bounds, four bins were used, corresponding to the intervals. This results in 28 features.

3.2. Edges

The next set of features is related to the existence of directed changes in intensity located near the pixel, so-called *edges.*

3.2.1. Sobel edge detection.

The Sobel filter performs a measure on the approximate spatial gradient of the grey intensity at the centre pixel. The following convolution kernels are computed:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

where A is the original image (in grey). The responses are combined using

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

The 2 features used were the response of the central pixel, and the mean response of the surrounding window.

3.2.2. Gabor filters.

The Gabor filter [5], like Sobel edge detection, is an approximate measure of spatial gradient at the centre pixel, but can be tuned for response to a wider variety of directions and frequencies. The convolutional kernel is computed by a Gaussian multiplied by a harmonic function. By distinguishing some predefined orientations and frequencies, a bank of kernels was obtained. We use the following formula

$$g(x, y; \lambda, \theta, \omega, \sigma, \gamma) = \exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) \cos(2\pi\frac{x'}{\lambda} + \omega)$$

$$x' = x\cos\theta + y\sin\theta$$

$$y' = -x\sin\theta + y\cos\theta$$

and defined 20 Gabor filters by using wavelengths $\lambda \in 3.5, 4.2, 5.0, 6.0, 7.4$ and rotations $\theta \in 0, 0.25\pi, 0.5\pi, 0.75\pi$. Further, $\omega = 0.5$, $\sigma = 0.35\lambda$ and $\gamma = 0$. These values have been derived by Hubel and Wiesel [6] as compatible with biological vision, and have been found to be good values in practical applications [7].

The computational complexity of computing the response for larger frequencies is substantial. A practical remedy, that reduces computations by an order of magnitude, is the following:

1. shrink the whole image by ratio α;
2. convolute the image by an equally shrunken kernel;
3. enlarge the convoluted image by ratio α.

After the convoluted images were calculated, the following features were defined for each pixel: maximum response over surrounding window, average response over surrounding window and the rotation of maximum response.

3.3. Texture

For many applications of pixel annotation, specifically for example for the Tarmac task, information related to the texture may be relevant. One way of recognizing textures is by means of a library of know textures. However, in most of our applications, we can safely assume a certain level of randomness, which forces us to use features that describe more general properties of the local texture. The following features were used.

3.3.1. Discrete Cosine Transformation.

One way to describe the texture of a surface is by using the *Discrete Cosine Transformation* (DCT). This transformation describes a function (or a signal) by the sum of cosines with different frequenties and amplitudes.

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}(n + \frac{1}{2})k\right] \quad k = 0, \dots, N-1$$

where N is number of pixels in the transformation input vector. According to Chiang and Knoblock [8], using a 2-dimentional DCT is an effective feature in classifying texture characteristics. The DCT function we implemented is equivalent to the transformation used by Chiang and Knoblock.

$$X_{k_1,k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \cos\left[\frac{\pi}{N_1}(n_1 + \frac{1}{2})k_1\right] \cdot \cos\left[\frac{\pi}{N_2}(n_2 + \frac{1}{2})k_2\right]$$

In this case $N_1 \times N_2$ describes the size of a 2-dimentional input vector around the pixel, for which the DCT is calculated. k_1 and k_2 are constructed in the same way as in the one-dimensional case, described earlier in this subsection.

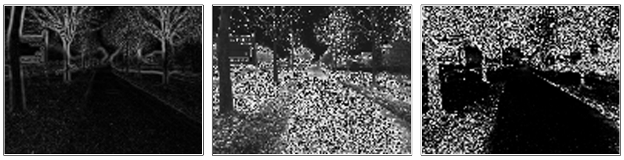


Figure 2. The image from Fig.1. after applying the Tamura features Coarseness (left), Contrast (middle) and Kurtosis (right).

3.3.2. Tamura.

As described by Howarth and Rüger, Tamura *et al.* tried to construct a collection of texture features that correspond to the human visual perception [9]. Howarth and Rüger compared the six features constructed by Tamura, by psychological measurements.

Coarseness is defined as the most powerful feature, introduced by Tamura et al. Images contain textures of several scales. The coarseness feature aims to identify the largest scale of the texture. This is done by taking the average over all *grey*-values for a number of neighbourhoods, and calculating the difference between two pairs of non-overlapping neighbourhoods, each on opposite sides of the pixel to classify. The average pixel value in a neighbourhood is calculated using the following equation:

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}} \sum_{j=y-2^{k-1}}^{y+2^{k-1}} I(i,j)/2^{2k}$$

where $I(i,j)$ is the *grey*-value of the image at coordinate (i,j) and k is the size of the neighbourhood. All neighbourhoods are of size $2^k \times 2^k$.

The difference between two sets of neighbourhoods is calculated in the following way:

$$E_{k,h} = |A_k(x-2^{k-1},y) - A_k(x+2^{k-1},y)|$$
$$E_{k,v} = |A_k(x,y-2^{k-1}) - A_k(x,y+2^{k-1})|$$

Here, the k value that maximizes the maximum value of $E_{k,h}$ and $E_{k,h}$ is used as input for the model.

Besides the *Coarseness* feature, we added the Tamura *Contrast* feature, as described by Howarth and Rüger [9]. This Contrast feature aims to capture the dynamic range of grey levels in an image, together with the distribution of black and white. For calculating the dynamic range of grey levels, the standard deviation is used, and the *kurtosis* α_4 is used to calculate the distribution of black and white. Combining both parts gives us the following *Contrast* measure:

$$F_{contrast} = \sigma/(\alpha_4)^n$$

In this formula the kurtosis α_4 is calculated by dividing the fourth moment around the mean *grey*-value by its variance squared. As our final texture feature we use this definition of *Kurtosis*:

$$\alpha_4 = \frac{(I(i,j)-\mu)^4}{\sigma^4}$$

Here, μ is the mean *grey*-value of the image and σ is the standard deviation.

The effect of the three Tamura features on the test image (Fig 1) is demonstrated in Fig. 2. As can be seen, especially the Kurtosis feature makes sense for the recognition of road surface.

3.4. 3D Information

As explained in the introduction, we use the Computer Vision package Harmonii for extracting 3D information from stereo images. For each pixel in the image, the 3-dimensional world coordinates are calculated, as well as a scalar reliability measure. The reliability measure is used to reduce the feature's sensitivity to noise, e.g. by taking a weighted mean instead of the true mean.

3.4.1. Height and Depth.

For the centre pixel, the height and depth were taken as features. From the surrounding window, the mean and variance were calculated for height and depth values. These features were calculated for the standard window size, and twice the standard window size, resulting in eight features per centre pixel.

3.4.2. Fitted Vertical Plane.

Least-squares linear regression was used to fit a vertical plane to the window, using

$$\hat{\beta} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

where $\bar{x}$ and $\bar{y}$ are the average x and y coordinates, and $\hat{\beta}$ is the optimal slope. The MSE(*mean squared error*) of the model is calculated as the variance from the model. Exposed to a window pointed at a vertical surface, such as a wall or a traffic sign, this method should find the correct slope, and variance from this slope should be low, relatively to non-vertical slopes.

4. Experiments

As mentioned, the Harmonii system was used to compute the different features from the stereo images and produce the necessary data files. As our classifier, we have selected the popular decision tree algorithm C4.5, as implemented by the J48 component of the Weka [10] learning environment built by the University of Waikato (New Zealand). Because Weka classifiers are programmed in Java, the resulting classifier could be easily integrated into Harmonii. Although C4.5 is a fairly standard algorithm, with some known limitations, it is useful in our setting because of its ease of interpretability. We like to stress that, depending on the specific pixel annotation task at hand, other classifiers may be a better choice. In our implementation, there are no limitations for using alternative classifiers from the Weka package.

4.1. Input

For each task, a number of (stereo) images was selected that are typical for the task at hand. Each stereo image consists of a left and right image, as well as a binary annotation image, as demonstrated in Fig. 1. The features related to the 3-dimensional information used both stereo images. For the intensity and colour-related features, only the left image was used. Each image was of resolution 640×480, which in theory produces $307,200$ data points per image. For reasons of efficiency, only every 5th pixel was computed and exported to file. Note that although not all pixels were computed, the neighbourhood and texture information still was computed on the highest possible resolution. Together with boundary pixels being ignored this resulted in $122 \times 90 = 10,980$ data points per image. Using Harmonii, all features were computed and exported to the *ARFF*-data files required by WEKA. There were 7 images available for the Tarmac task, and 5 for the Sign task, which resulted in two datasets of sizes $76,860$ and $54,900$. In the Tarmac dataset, 34.97% was assigned positive, and in the Sign dataset 3.02%.

As cross-validation approach, we have opted for a strategy that is somewhat different from the usual approach. Rather than dividing the dataset in n random subsets of equal size, we divide according to the source image. This means that a classifier is trained on the data resulting from $n-1$ images and then tested on the data from the remaining image. This process is then repeated for each image, and results are averaged. As such, we do not just test the normal generalisation capabilities of the classifier, but also the generalisation from one situation to the next. Note that there may be many changes between images depending on the weather and lighting conditions. We require our classifiers to deal with such variations.

Table 1. Results (in % and standard deviations between brackets) on the two different task.

	without 3D	with 3D
Tarmac	92.0 (± 5.3)	92.3 (± 6.0)
Sign	99.5 (± 0.147)	99.5 (± 0.147)

4.2. Results

The results of building and cross-validating the classifier on the two task are shown in Table 1. An immediate conclusion from this table is that both tasks show good performance. The Tarmac task shows a considerable increase, from a baseline of 65.03% to 92.3% (with all features included). The Sign tasks shows a near perfect 99.5% on average, but we have to take into account a baseline of 96.98% (signs are relatively small and do not appear that often). Still the increase is considerable, given that highly skewed target distributions are notoriously hard in machine learning.

As can be seen from the table, the effect of adding 3D features has at most a marginal effect. In the Sign task, no clear effect could be observed. It should be noted that this effect is not so much due to the unimportance or unreliability of the 3D features, but rather to redundancy in the large set of features produced. In the case of Tarmac, most pixels below the horizon actually belong to the ground. This means that both the 2D information (where in the image?) as well as the 3D information (where in the outside world?) is equally informative. Only where obstacles appear below the horizon, but are not directly on the ground, the 3D features may be more predictive. In fact, in the Tarmac task, the 3D features do appear in decision trees, just not always as the first split. For the Sign task, the classifiers are quite simple, and only use the saturation of the colour, and the 2D information. We would like to stress that for other tasks, the 3D features may become more important than any of the other features, for example when the distance from the observer is relevant. This is information that in general can only be obtained using stereo vision.

The colour and 2D information appear to be important in both selected tasks. In the Tarmac task, the 2D information was most relevant, followed by colour and 3D information. Finally, texture information was least important in this task. As mentioned, colour was most relevant for the Sign task, followed by 2D information. In all cases, almost equally performing classifiers can be obtained by removing some of the features used, hinting to the level of redundancy amongst the features. On the whole, the edge features did not appear

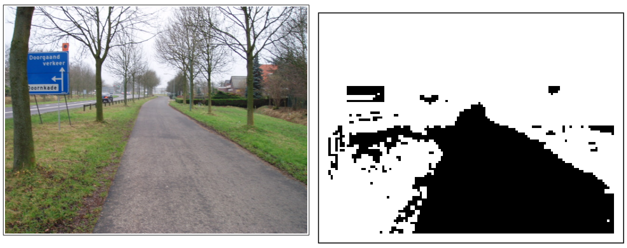


Figure 3. The results of applying a classifier to one of the images in the Tarmac task.

to be very informative.

By informal inspection of the classified images, a sense can be obtained of the types of mistakes the classifiers tend to make. On the whole, false positives are more frequent than false negatives. This means that, for example, areas outside the actual road may sometimes be labelled 'road', but on the road itself, very few mistakes are made. Often, the miss-classifications appear in areas where the annotation is arguable anyhow, such as pavements that are made of material very similar to tarmac. Surprisingly, the classifiers seem to have little trouble with complications such as road markings or shadows on the road, nor did the classifier struggle with the multiple colours appearing on the signs.

4.3. Path Planning in Autonomous Vehicles

As a simple demonstration of how pixel annotation can aid larger tasks based on Computer Vision, we show how the tarmac-identification classifier can be employed in an automotive setting. In Fig. 3, we see a scene with a road and a number of obstacles, as observed from an autonomous vehicle (left picture). Additionally, we see where a trained classifier thinks areas of road surface are (right picture). Purely based on 3D information, the navigation system (included in Harmonii) would identify the relevant obstacles, in this case three trees, and suggest any path that does not lead to collisions. In this case however, following the road is more urgent than avoiding obstacles. A straightforward solution to this is to simply look up the 3D information for each non-tarmac pixel in the classified image, and treat these as possible obstacles also.

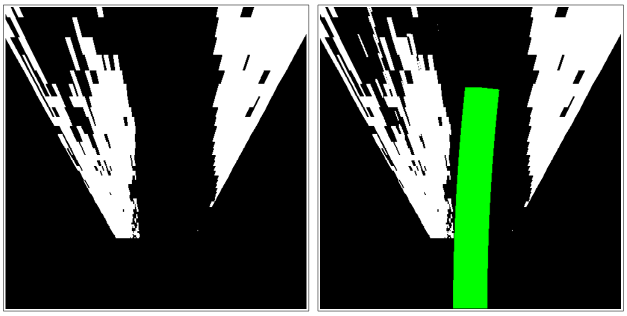


Figure 4. The results of applying the inverse perspective tranformation to the classified image, and planning a trajectory for a mobile robot.

As a more challenging setting, assume we have only a single camera, and thus no depth information. On first sight, it seems that we have insufficient information to follow the road, as we do have reliable classification, but no way to translate this into 3D information. However, by adding a simple assumption, we can obtain this information. If the vehicle is situated on (or near) the road, and can identify this road, then we can safely assume that the road is in a horizontal plane at some measurable distance below the camera. This means that with a simple transformation of the 2D image, we can map the identified tarmac on the horizontal plane. This operation is known as the *inverse perspective* transformation [11], and works as follows:

$$x = \frac{g \cdot x'}{y'}$$

$$y = g$$

$$z = \frac{c \cdot g}{y'}$$

where x, y and z are the induced 3D coordinates, x' and y' are the coordinates in the image (origin in the middle of the image on the horizon). The camera is assumed to be positioned horizontally. c is a constant related to the field of view and resolution of the camera. g is the measured height from the ground in meters.

The effect of this transformation is displayed in Fig. 4 (left). Note that the inverse perspective basically skews and stretches the original image. In Fig. 4 on the right the effect of simply navigating on the classified and transformed tarmac image is demonstrated. Clearly, the system suggests a path in 3D that is based on information from a single camera.

5. Conclusion

We have presented a method for pixel annotation: classifying regions in images into predefined discrete classes, such as 'road' and 'non-road'. Our method relies heavily on a collection of features extracted from the original image(s). These features capture different properties of the pixel itself, but also of the context it appears in (built up by the neighbouring pixels). These features can be divided into a number of groups: colour, edge, texture and location features. For the last group, we have 2D information concerning the location of the pixel in the image, as well as

3D information about the likely location of the pixel in the outside world. This 3D information was obtained by using stereo images, and computing the 3D information in the Harmonii Computer Vision system [2].

Experiments show that, even with a straightforward classifier, reliable results can be obtained for realistic pixel annotation tasks. We have tested two tasks in an automotive setting and obtained good results in both. This is not trivial, as the classifier has to deal with a lot of complexities of real-life situations. These include natural variations in the situation itself, but also in the lighting and weather conditions. By choosing a number of images that show these variations, robust classifiers could be obtained. It turns out that most of the mistakes made by the classifiers are logical: the actual annotation is ambiguous, or humans would also have a hard time making correct predictions without domain knowledge or using context information.

As was demonstrated in Section 4.3, the pixel annotation module can be a useful component in larger systems, such as autonomous vehicles. The task of navigating along a road can be facilitated by simply recognising the road-surface, and suggesting a path that respects this surface. Note that even without having actual 3D information, for example because only a single camera is available, some spatial reasoning can be done using the classifier's output (see Fig. 4). It turns out that the set of features used is overcomplete, and if certain information is lacking (e.g. spatial) other features may reliably take their place.

References

[1] Alvy Ray Smith. Color gamut transform pairs. *Computer Graphics*, 12 (3), 1978.

[2] Harmonii. `http://www.kiminkii.com/harmonii.html`.

[3] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 34-40. IEEE, June 2005.*

[4] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 3, March 2007.*

[5] P. Kruizinga and N. Petkov. Non-linear operator for oriented texture. *IEEE Trans. on Image Processing*, 8(10):1395–1407, 1999.

[6] D. Hubel and T. Wiesel. Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *J Neurophysiol*, 28:229–89, 1965.

[7] Stanley Bileschi and Lior Wolf. *A Unified System For Object Detection, Texture Recognition, and Context Analysis Baed on the Standard Model Feature Set.* The Center for Biological and Computational Learning, Massachusetts Institute of Technology, 2002.

[8] Yao-Yi Chiang and Craig A. Knoblock. Classification of line and character pixels on raster maps using discrete cosine transformation coefficients and support vector machine. *ICPR - Proceedings of the 18th International Conference on Pattern Recognition*, 2:1034 – 1037, 2006.

[9] P. Howarth and S. Rüger. Robust texture features for still-image retrieval. *Vision, Image and Signal Processing, IEE Proceedings*, 152:868 – 874, 2005.

[10] Weka. `http://www.cs.waikato.ac.nz/~ml/weka/`.

[11] T. Schouten and E. van den Broek. Inverse perspective transformation for video surveillance. *Computational Imaging VI : proceedings of SPIE, vol. 6814. SPIE, the International Society for Optical Engineering*, 2008.

Extended Abstracts

Opti-Fox: Towards the Automatic Tuning of Cochlear Implants

Keywords: cochlear implants, tuning, fitting, modelling, optimisation

Zoltán Szlávik* Z.SZLAVIK@VU.NL
*VU University Amsterdam, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands

Bart Vaerenberg** VAERENBERG@EARGROUP.NET
**the Eargroup, Herentalsebaan 71, B-2100 Antwerp-Deurne, Belgium

Wojtek Kowalczyk* W.J.KOWALCZYK@VU.NL
Paul Govaerts** DR.GOVAERTS@EARGROUP.NET

Abstract

This paper describes the Opti-Fox project which aims at reducing the need for the presence of experts when fitting cochlear implants to patients.

1. Introduction

Hearing is a complex sensory function which converts mechanical waves (sound) to electrical patterns on the auditory nerve. The sound receptor is the inner ear or the cochlea. Each year Europe counts more than 5000 deaf-born babies (i.e., 1/1500 newborns). Not treating these babies leads to deaf-muteness. In adults, the prevalence of "cochlear" hearing loss increases with age, and approximately 20% of 75-year old people have a moderate, and 3% a severe hearing loss. Severe and profound hearing losses can be treated with cochlear implants (CI). When implanting a CI, some 10 to 20 contacts are surgically placed into the cochlea. Within the device, sound is analysed by the external "speech processor" which resembles a classical behind-the-ear hearing aid, and implanted electrodes are then stimulated to generate an electrical field to pass the information directly to the auditory nerve.

The working of a CI is controlled by about 200 tunable parameters that determine the "input-output" behaviour of the speech processor: sensitivity levels at different frequencies, electrical dynamic ranges for each electrode, characteristics of amplifiers, and strategies of stimulating electrodes.

After implantation, CIs need to be programmed or "fitted" to optimise the hearing sensation of the individual patient (Clark, 2003). This is a challenging and time-consuming task that is typically performed by highly trained engineers, audiologists or medical doctors. One of the reasons for this is that the outcome of the optimization process is difficult to measure. Tone and speech audiometry are the only outcome measures that are clinically used all over the world. However, they only measure a marginal aspect of the entire auditory performance, and they hardly allow analytical feedback to the fitter. Therefore, many fitters mainly rely on the patient's instantaneous feedback. However, since many patients are very young, or may never have been hearing "normally" before, this feedback often relates more to comfort than to intrinsic sound coding accuracy.

As a result, CI centres and manufacturers have developed their own heuristics, usually in the form of simple "if-then" rules that are applied in a very flexible, individual and uncontrollable way. At present, no universal standards or well defined Good Clinical Practice exists to guide the fitters. With more than 200000 CI users worldwide and an annual increase of over 30.000, this becomes an ever-increasing problem and a real bottleneck to further implementation.

2. Opti-Fox

Opti-Fox is a project which is funded by a European Commission's 7th Framework Research for SME grant (7FP-SME 262266). In this project we aim at developing an intelligent, self-learning agent (or system) for CI fitting. We will combine the latest technologies from linguistics, automatic speech recognition, machine learning, and optimization techniques. The con-

Appearing in *Proceedings of the 20^{th} Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the author(s)/owner(s).

sortium consists of SMEs and research institutes from Belgium, the Netherlands and Germany, in close collaboration with the CI manufacturer Advanced Bionics. For more details, the reader is referred to the project's website[1].

2.1. New tests to measure fitting quality

Ongoing research focuses on the development of a new speech understanding test that is language independent, provides automatic scoring by means of Automatic Speech Recognition (ASR) technology, and allows detailed spectral analysis to feed back to the CI fitter. Together with linguists and computational scientists we are developing bin definitions that characterise languages. They will be validated on large corpora of several European languages and they will allow us to draw customised samples for individual patients in such a way that the samples are representative of the entire language in terms of phonetics, typology, morpho-syntax, etc.

We are also developing automated scoring strategies by means of ASR. Different algorithms are being developed and investigated to compare reference utterances with test utterances which may contain errors due to the test person's hearing deficit. In a first stage we have made recordings in 30 hearing volunteers with Flemish, Dutch or German as mother tongue, who were asked to (re)produce about 400 words twenty times, yielding around 240.000 wave-files. This vast dataset will serve the first validation of the algorithms under development. Next stages will include the validation in hearing impaired subjects.

2.2. Modelling

By analysing all (both historical and the most recent) data the system will be learning the model of a complex relation between settings of CI device parameters and hearing performance. Our aim is to develop a system that is able to recommend adjustments to device parameters that will lead to the maximal improvement of hearing quality.

The main challenges in modelling, and, subsequently, in tuning a CI device's parameters arise from the following facts: As already mentioned, the number of parameters to be tuned is high ($\sim$200) compared to the number of records available in the data collected so far (which is in the order of several thousands). In addition to the parameters of the CI device, one should also consider the differences among patients (e.g., age, speech intelligibility, severity of hearing loss, etc.). Also, producing new records is a time consuming and expensive task (obtaining one set of exhaustive test results can take up to two hours). An exhaustive hearing test consists of several types of tests (e.g., Audiogram, Phoneme tests), which have several sub-tests (e.g., Audiogram at 1000Hz, 2000Hz, etc.). Currently, the number of all sub-tests is more than fifty. Due to their expensive nature, and sometimes due to assumptions made by experts, some of these tests are not even carried out. Missing values are also routinely present in the input data. In addition, the types and ranges of outcome values are different, e.g., Audiogram measures performance in decibels, values typically ranging from 0 to around 70 for each of the 8 frequencies at which measurement is taken, while a Phoneme score is the sum of 20 binary sub-tests. The various (often asymetric) error measures that can be defined for (sub-)tests to evaluate model performance complicates modelling even further.

2.3. Optimal Tuning

The model provides only a probabilistic approximation of the expected response of a patient to particular parameter settings. Nevertheless, it can be used to compute the most promising configuration that should be tried next. After the evaluation of the recommended configuration by testing the patient, the model could be updated to become more “patient-specific” and, therefore, accurate. Consequently, the updated model can be used to find the next most promising configuration, and so on, until a satisfactory configuration of the CI is found.

3. Concluding remarks

An intelligent agent, called FOX, that facilitates CI fitting has been developed, CE marked and is already in use in several CI-centres across Europe (Vaerenberg et al., 2011). The first prototype system of Opti-Fox is expected to be finished around the end of 2011, while the commercial version of the system will be available at the end of 2012.

References

Clark, G. M. (2003). *Cochlear implants : fundamentals and applications*, 663–670, 679–683. Springer, New York.

Vaerenberg, B., Govaerts, P. J., de Ceulaer, G., Daemers, K., & Schauwers, K. (2011). Experiences of the use of FOX, an intelligent agent, for programming cochlear implant sound processors in new users. *International Journal of Audiology*, *50*, 50–58.

[1] http://www.otoconsult.com/opti-fox

Determining the Diameter of Small World Networks

Frank W. Takes and Walter A. Kosters {FTAKES,KOSTERS}@LIACS.NL
Leiden Institute of Advanced Computer Science (LIACS), Leiden University

1. Introduction

In this ongoing work we study the *diameter*: the length of the longest shortest path in a graph. We specifically look at the diameter of *small world networks*: sparse networks in which the average distance between any pair of nodes grows only proportionally to the logarithm of the total number of nodes in the network.

The diameter is a relevant property of a small world network for many reasons. For example in social networks (Mislove et al., 2007), the diameter could be an indication of how quickly information reaches everyone in the network. In an internet topology network, the diameter could reveal something about the worst-case response time between any two machines in the network. Other small world networks include scientific collaboration networks, gene networks, and the web.

Exact algorithms for calculating the diameter of a graph traditionally require running an *All Pairs Shortest Path* (APSP) algorithm for each node in the network, ultimately returning the length of the longest shortest path that was found. Sadly, with n vertices and m edges, complexity is in the order $O(n^3)$ for general weighted graphs and $O(mn)$ for sparse unweighted graphs such as our small world networks. The naive APSP algorithm for obtaining the diameter is clearly not feasible in large graphs with for example millions of nodes and a billion links, which are common numbers for structured datasets that are nowadays studied.

Whereas previous work focuses on approximating the diameter, for example based on a sample, we propose an algorithm which can determine the *exact* diameter of a small world network by taking advantage of the characteristic properties of such networks. Our algorithm uses lower and upper bounds and attempts to evaluate only the critical nodes which ultimately determine the diameter. Experiments show that our algorithm greatly improves upon the APSP algorithm.

2. Definitions & Observations

Given an undirected graph $G(V, E)$ with n vertices (or nodes) and m edges and a distance function $d(v, w) = d(w, v)$ which computes the length of the shortest path between $v, w \in V$, we can define the *eccentricity* of a node $v \in V$, written as $\varepsilon(v)$, to be $\max_{w \in V} d(v, w)$: the length of the longest shortest path starting at node v. We will denote the *diameter* of a graph G by $\Delta(G)$, and define it as $\max_{v,w \in V} d(v, w)$, or equivalently as the maximum eccentricity over all nodes: $\max_{v \in V} \varepsilon(v)$.

If we calculate the eccentricity $\varepsilon(v)$ for some node v, we know that for all nodes w with $d(w, v) = k$, their eccentricity $\varepsilon(w)$ lies between $\varepsilon(v) - k$ and $\varepsilon(v) + k$. The upper bound follows from the fact that any node w at distance k of v can get to v in exactly k steps, and then reach any other node in at most $\varepsilon(v)$ steps. The lower bound can be derived by interchanging v and w in the previous statement. Furthermore, this lower bound can of course never be less than k, because a path of length k to w apparently exists. More formally:

> If a node v has eccentricity $\varepsilon(v)$, then for all nodes $w \in V$ we have $\max(\varepsilon(v) - k, k) \leq \varepsilon(w) \leq \varepsilon(v) + k$, where $k = d(w, v)$.

We know that the diameter of a graph is equal to the maximum eccentricity over all nodes. Therefore, the maximum lower and upper bound bound on the eccentricity over all nodes, can be seen as a lower and upper bound on the diameter. Also, because the graph is undirected, the upper bound on the diameter of the graph can never be more than twice as big as the smallest eccentricity upper bound over all nodes. These observations can be formalized as follows:

> Let $\varepsilon_L(v)$ and $\varepsilon_U(v)$ denote currently known lower and upper bounds for the eccentricity of node $v \in V$. For the diameter $\Delta(G)$ of the graph it holds that $\max_{v \in V} \varepsilon_L(v) \leq \Delta(G) \leq \min(\max_{v \in V} \varepsilon_U(v), 2 * \min_{v \in V} \varepsilon_U(v))$.

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).
This work is currenty under review elsewhere as full paper.

3. Algorithm

Whereas the APSP algorithm will calculate the eccentricity of all n nodes in the network in order to obtain the diameter, our goal is to only calculate the eccentricity of certain vital nodes that contribute to the diameter. The process is outlined in Algorithm 1. In essence, we are repeatedly selecting a node (line 7), calculating its eccentricity (line 9), and updating the bounds of the diameter (Δ_L and Δ_U, lines 10–11) and node eccentricities (lines 13–14) according to the two observations from the previous section. In this process we hope to quickly disregard many nodes, which can happen either because the lower and upper eccentricity bounds of a node can no longer contribute to the diameter bounds, or because the exact eccentricity of that node is already known because its lower and upper eccentricity bounds have become equal (lines 15–17).

Figure 1 shows the lower and upper eccentricity bounds (beneath and above nodes, respectively) after calculating the eccentricity of node F in the first iteration of the algorithm. One may verify that after calculating the eccentricity for T and L, all other nodes can be discarded and the actual diameter has been found.

The selection strategy (line 8) for selecting the next node for which we are going to calculate the eccentricity is clearly vital. Experiments show that the best results are obtained when we repeatedly interchange the selection of the node with the largest upper bound and the node with the smallest lower bound. This can be seen as a repeated attempt to increase the the lower bounds and decrease the upper bounds.

Algorithm 1 BoundingDiameters

```
1: Input: Graph $G(V,E)$        Output: Diameter of $G$
2: $W \leftarrow V$;  $\Delta_L \leftarrow -\infty$;  $\Delta_U \leftarrow +\infty$;
3: for $w \in W$ do
4:     $\varepsilon_L[w] \leftarrow +\infty$;  $\varepsilon_U[w] \leftarrow -\infty$;
5: end for
6: while ($\Delta_L \neq \Delta_U$ and $|W| > 0$) do
7:     $v \leftarrow$ selectFrom($W$);
8:     $W \leftarrow W - \{v\}$;
9:     $\varepsilon[v] = $ eccentricity($v$);
10:    $\Delta_L = \max(\Delta_L, \varepsilon[v])$;
11:    $\Delta_U = \min(\Delta_U, 2 * \varepsilon[v])$;
12:    for $w \in W$ do
13:        $\varepsilon_L[w] = \max(\varepsilon_L[w], \max(\varepsilon[v] - d(w,v), d(w,v)))$;
14:        $\varepsilon_U[w] = \min(\varepsilon_U[w], \varepsilon[v] + d(w,v))$;
15:        if ($\varepsilon_U[w] \leq \Delta_L$ and $\varepsilon_L[w] \geq \Delta_U/2$) or
               ($\varepsilon_L[w] = \varepsilon_U[w]$) then
16:            $W \leftarrow W - \{w\}$;
17:        end if
18:    end for
19: end while
20: return $\Delta_L$;
```

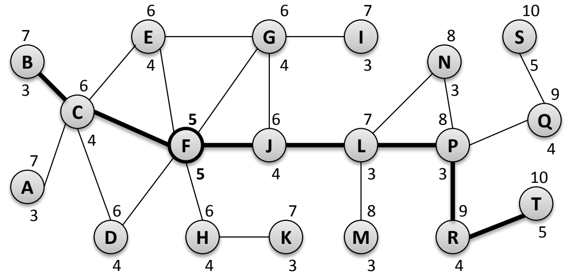


Figure 1. A sparse graph with 19 nodes and 23 edges. The path B-T realizes the diameter (length 7).

4. Results and Conclusion

The results of testing our algorithm on the undirected versions of the connected component of various small world datasets are given in Table 1. The number of nodes is equal to the number of eccentricity calculations in the column “APSP”. Results show that our algorithm (column “BD”) is able to calculate the exact diameter Δ with only a fraction of the calculations of the APSP algorithm.

Dataset	Δ	Edges	APSP	BD
AstroPhys	14	396,160	17,903	9
Hyves	25	912,120,070	8,057,981	21
LiveJournal	23	97,419,546	5,189,809	3
Orkut	10	234,370,166	3,072,441	106
InternetTopo	31	22,190,596	1,696,415	4

Table 1. APSP vs. BoundingDiameters (BD). Datasets courtesy of (Leskovec et al., 2007), Hyves (anonymized), (Mislove et al., 2007) and (Leskovec et al., 2005).

In future work we would like to investigate if our algorithm can determine the eccentricity of all nodes in order to study the exact eccentricity distribution. We also hope to investigate how the exact diameter of small world networks behaves over time.

Acknowledgments This research is part of the NWO COMPASS project (grant #612.065.92).

References

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2005). Graphs over time: densification laws, shrinking diameters and possible explanations. *Proceedings of KDD2005* (pp. 177–187).

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2007). Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, *1*, article 2.

Mislove, A., Marcon, M., Gummadi, K., Druschel, P., & Bhattacharjee, B. (2007). Measurement and analysis of online social networks. *Proceedings of the 7th SIGCOMM Conference on Internet Measurement* (pp. 29–42).

Automatic Construction of Personalized Concept Maps from Free Text

Zoltán Szlávik Z.SZLAVIK@VU.NL
Pedro Nogueira PBA260@FEW.VU.NL
VU University Amsterdam, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands

Keywords: concept maps, personalization, free text, learning

Abstract

This paper presents an ongoing effort to create a software application to automatically create concept maps from free text. The goal is not to create human-quality concept maps, but rather to jump-start their creation, for later refinement by a person. The application will be able to learn from the refinements and parameters set by the user, to improve future concept map constructions.

1. Introduction

A concept map is a useful schematic resource used to represent and organize a set of concepts in a propositional structure. Concept maps were developed by Novak in 1972 and were used to identify changes in children's understanding of scientific knowledge (Novak & Musonda, 1991). This research, in turn, was based on Ausubel's theory that the learning process is based on the assimilation of concepts into knowledge frameworks already held by the learner (Ausubel, 1978). This implies and emphasizes the point that learning is not a bucket waiting to be filled but a system of interrelated memory systems interacting with various inputs. Learning utilizes working memory, which can only process a limited amount of information units; as such, concept maps are useful to group information units into concepts such that more information may be processed. Concept maps are useful tools in aiding this interaction of memory systems and support the recognition of valid and invalid ideas.

In general, a concept map requires much iteration to be improved. When the goal is to build knowledge models, especially for large domains, the cost of determining which concepts and relations to choose is a tough task, even though this effort carries its own benefits by allowing students to increase their own understanding by performing the knowledge modeling process. Considering these issues, computer programs may work in favor of the practice of thinking using concept maps.

The present paper presents an ongoing effort to create a software application to automatically construct a concept map from natural language text. A number of information extraction procedures are used to create a preliminary version of the concept map, which will work as a starting point for humans to adapt. The goal is not to create a human-quality concept map, but rather to jump-start its creation, for later refinement by a person. The application will then learn from the refinements to improve future concept map constructions.

2. Triplet Extraction

The core of the application consists of the triplet extraction algorithm. In this context, a triplet is defined as a relation between subject and object, where the relation is the predicate. The goal is, then, to extract sets of the form {subject, predicate, object} out of the syntactically parsed sentences.

We make use of the Stanford typed dependencies representation to extract the set of grammatical relations for each sentence (Marneffe & Manning, 2008). A dependency defines a binary relation between a *governor* and a *dependent*.

The extraction method consists of two basic steps. First, a set of noun phrases is extracted, which will become the set of candidate concepts. Any modifier accompanying a noun phrase is attached to the candidate concept. Then, all pairs of concepts that have a dependency link between them through a verb phrase are extracted. The verb phrase shows the relation between the concepts and so it is used as the linking phrase to form the triplet.

Once the extraction steps are done, we perform co-reference resolution to match concepts that relate to the same entity. Finally, the triplets are merged to form the building blocks of the semantic graph, where concepts are nodes and the relation between them is labeled with the linking phrase. Figure 1 shows an example output of this algorithm.

3. Parameter Settings and Learning

The user is encouraged to tweak the concept map to find a better representation of the knowledge at hand. With these refinements the application will improve

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the author(s)/owner(s).

The atom is a basic unit of matter that consists of a dense, central nucleus surrounded by a cloud of negatively charged electrons. The atomic nucleus contains a mix of positively charged protons and electrically neutral neutrons.

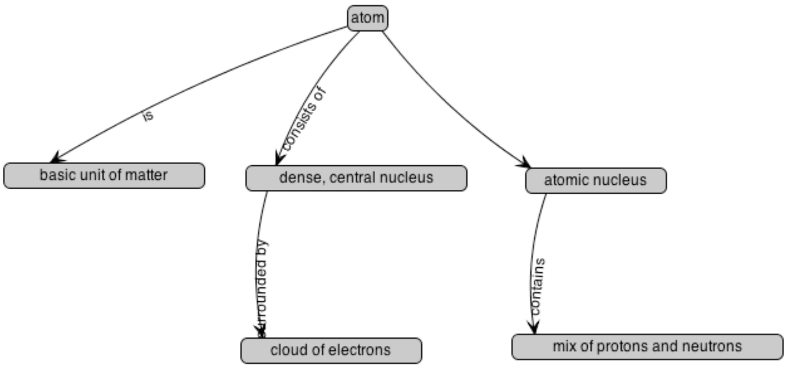


Figure 1. Example output of the algorithm (right) for a short document (left).

future creations of concept maps by learning how to arrange the concepts in a way that is closer to a user's liking. The application offers three ways by which this can be done:

- Concept ranking – Assigns weights to individual concepts. Algorithms for concept ranking include setting the weights according to the number of occurrences of a given concept in the text, or according to the number of different relations a concept is involved in. Each algorithm comes with a number of parameters that can be adjusted, and the application can easily be extended to include more ranking measures.
- Layout – Sets the graphical representation of the concept map. A concept map would normally be oriented top-down with higher-ranking concepts higher up on the map and more specific concepts below. However, a user might prefer a presentation where the concepts are spread outward from the center. These preferences will be incorporated in future concept maps. Note that it is not only the style of the layout that can be learned and used later, but also, position of concepts might also provide clues as for the generation of subsequent concept maps. As with concept ranking, the application can easily be extended to include more layouts.
- Add/remove concepts and relations – The user may add new concepts and relations, or remove existing ones, which will further allow the refinement of the algorithms involved in concept map generation.

All the changes to the concept map done by the ways described above are translated into parameter adjustments of the map-generating algorithm. Not only will we be able to learn a personalized user profile, but also generally good map generation and display settings, so we can improve the quality of 'first maps' (or baseline maps) presented to users who are new to the application.

The features for learning that we propose exist on two levels: a *linguistic* level, where we check whether concepts affected by user actions are in subject or object role, and what the part-of-speech tags of the words forming the concept are. At the *document* level, we check for the location of the triplets in the document, and the frequency and location of words inside the sentences.

4. Conclusions

In this paper we presented an ongoing effort in building an application to automatically construct concept maps from natural language text. Furthermore, we proposed the application of machine learning techniques to better their construction. Even though the problem of the automatic construction of concept maps is not new (e.g.: Clariana & Koul, 2004; Rusu et al., 2009), we believe that the application of machine learning techniques to the refinements made by users in a tool with powerful editing capabilities is an interesting research direction.

References

Ausubel, D., Novak, J. D., & Hanesian, H. (1978). *Educational Psychology: A Cognitive View (2nd Ed.)*. New York: Holt, Rinehart & Winston.

Clariana, R. B., & Koul, R. (2004). *A Computer-based Approach for Translating Text into Concept map-like Representations*. Proceedings of the First International Conference on Concept Mapping.

Marneffe, M., & Manning, C. D. (2008). *The Stanford Typed Dependencies* Representation. COLING Workshop on Cross-framework and Cross-domain Parser Evaluation.

Novak, J. D., & Musonda D. (1991). *A twelve-year longitudinal study of science concept* learning. American Educational Research Journal, 28(1), 117-153.

Rusu, D., Fortuna, B., Grobelnik, M., & Mladenic, D. (2009). *Semantic Graphs Derived from Triplets with Application in Document Summarization*. Informatica Journal, 33, 357-362.

Multi-Objective Evolutionary Art

Keywords: Multi-Objective Optimization, Evolutionary Art

Eelco den Heijer EELCO@OBJECTIVATION.NL
Objectivation, Amsterdam, Netherlands
Vrije Universiteit Amsterdam, Netherlands

A.E. Eiben GUSZ@CS.VU.NL
Vrije Universiteit Amsterdam, Netherlands

Abstract

In this paper we investigate the applicability of Multi-Objective Optimization (MOO) in Evolutionary Art. We evolve images using an unsupervised evolutionary algorithm and use two aesthetic measures as fitness functions concurrently. We use three different pairs from a set of three aesthetic measures and compare the output of each pair to the output of other pairs, and to the output of experiments with a single aesthetic measure (non-MOO). We investigate 1) whether properties of aesthetic measures can be combined or merged using MOO and 2) whether the use of MOO in evolutionary art results in different, or perhaps even "better" images.

1. Introduction

This paper is an abstract of recent work (den Heijer & Eiben, 2011) in which we investigate whether it is possible to evolve aesthetic images by combining the effects of multiple aesthetic measures concurrently using a Multi-Objective Evolutionary Algorithm (MOEA). In previous work we have shown that the choice of the aesthetic measure significantly determines the "style" of the generated art (den Heijer & Eiben, 2010). With MOEA, we want to investigate whether the influence of different aesthetic measures can be combined or merged into the same image. Our first research question is; can we combine the effects from multiple aesthetic measures into the same image using a MOEA? Second, we want to know whether the use of a MOEA results in "better" images in evolutionary art.

2. MOEA and Evolutionary Art

Multi-Objective Evolutionary Algorithms (MOEA) are evolutionary algorithms that use multiple fitness functions (or objectives) to evolve solutions to certain problems. In our research we approach the evolution of art as an optimization problem, and use multiple aesthetic measures in order to combine different aspects of aesthetic evaluation concurrently. MOEA's have not been used frequently in the field of evolutionary art (den Heijer & Eiben, 2011). In our experiments we used three aesthetic measures as fitness functions. Aesthetic measures are functions that assign an aesthetic score to an image. The aesthetic measures that we use in our experiments are Benford Law, Global Contrast Factor and Ross & Ralph (den Heijer & Eiben, 2011).

3. Experiments and Results

We did a number of experiments to evaluate the use of a MOEA in evolutionary art. Our evolutionary art system uses genetic programming (den Heijer & Eiben, 2011). In all our experiments we used a population size of 200, 20 generations, a tournament size of 3, a crossover rate of 0.9 and a mutation rate of 0.1.
We performed three experiments with the well-known NSGA-II algorithm using two aesthetic measures of the following three: 1) Benford Law and Ross & Ralph, 2) Global Contrast Factor and Ross & Ralph and 3) Benford Law and Global Contrast Factor. We did 10 runs with each setup, using the exact same experimental setup except for the combination of aesthetic measures. From each run, we saved the Pareto front (the first front, with rank 0) and calculated the normalized

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).

fitness for image I for each objective f between 0 and 1. Next, we ordered each individual on the sum of the normalized scores of the two objectives, and we stored the top 3 individuals from each run. With 10 runs per experiments, we have 30 individuals per experiment that can be considered the "top 30". Using this approach, we have a fair and unbiased selection procedure (since we did not handpick images for these selections). In the top 30 portfolio of the experiment with Benford Law and Ross & Ralph (Figure 1) we can clearly see the influence of both aesthetic measures in the images. The Benford Law aesthetic measures produces images with an organic, natural feel and the Ross & Ralph measure tends to produce image with a "painterly" feel (since it focuses on smooth transitions in colours). We can see these properties in most images and in some images they are combined (i.e. in the first three images in Figure 1). The last two images of the second row and the first image of the third row also appear in the close-up of the Pareto front in Figure 2. For more details and images of the experiments with the other combinations of aesthetic measures we refer to (den Heijer & Eiben, 2011).

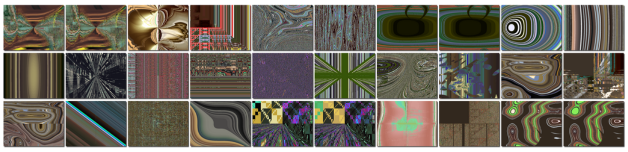


Figure 1. Portfolio of images gathered from ten runs with NSGA-II with Benford Law and Ross & Ralph

3.1. Close-ups of Pareto fronts

We wanted to know in detail how a single Pareto front was organized, and whether we could see a gradual transition of the influence of measure A to measure B while moving over the Pareto front. We zoomed in on a single Pareto front and reconstructed the images that belong with each individual in that front. In the following figure we show the Pareto front for each pair of aesthetic measure (note that we did 10 runs per experiments, but we only show the Pareto front of one run). In Figure 2 we see the 15 individuals plotted based on their scores on the Ross & Ralph measure and the Benford Law measure. We normalized the scores between 0 and 1.

If we look at the individuals of the Pareto front in Figure 2, we can see a transition of the influence from aesthetic measure to the other. At the top we see "typical" Ross & Ralph images (den Heijer & Eiben, 2010; den Heijer & Eiben, 2011), and at the bottom/ right we see more typical Benford Law images. In between, at the right/ top we see the images where the influences blend most.

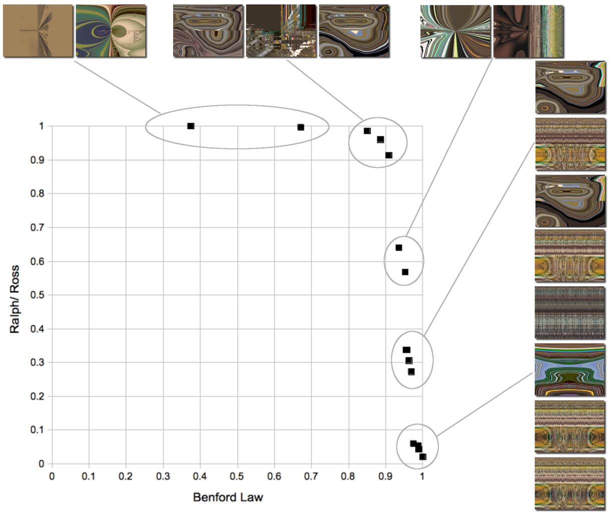


Figure 2. Details of the Pareto front of Benford Law and Ross & Ralph with the corresponding images per element of the front.

4. Conclusions and Future Work

Our first research question was whether the influence of different aesthetic measures could be combined into the evolved images. From our experiments we can conclude that we actually can, but that the combination of aesthetic measures should be chosen with care. Combinations of aesthetic measures that have opposing goals will result in inefficient search behaviour, and will not result in images with "synergy" of the aesthetic measures. Our experience shows that combinations of aesthetic measures that have "somewhat" different goals result in the most interesting images. Our second research question was whether the resulting evolved images are more interesting when using multiple aesthetic measures. If we compare the images of our MOEA experiments with images of previous work with a single aesthetic measures, we can conclude that the MOEA images are (on average) more interesting.

References

den Heijer, E., & Eiben, A. (2010). Comparing aesthetic measures for evolutionary art. *Applications of Evolutionary Computation, LNCS 6025, 2010* (pp. 311–320).

den Heijer, E., & Eiben, A. (2011). Evolving art using multiple aesthetic measures. *EvoApplications, LNCS 6625, 2011* (pp. 234–243).

Using Domain Similarity for Performance Estimation

Vincent Van Asch VINCENT.VANASCH@UA.AC.BE
Walter Daelemans WALTER.DAELEMANS@UA.AC.BE
CLiPS - University of Antwerp, Prinsstraat 13, 2000, Antwerp, Belgium

Abstract

This paper explores a number of measures that attempt to predict the cross-domain performance of an NLP tool through statistical inference. We apply different similarity measures to compare different domains and investigate the correlation between similarity and accuracy loss of a part-of-speech (POS) tagger. We find that the correlation between the performance of the tool and the similarity measure is linear and that the latter can therefore be used to predict the performance of the NLP tool on out-of-domain data.

1. Introduction

When developing an NLP tool using supervised learning, annotated data with the same linguistic properties as the data for which the tool is developed is needed, but not always available. In many cases, this means that the developer needs to collect and annotate data suited for the task. When this is not possible, it would be useful to have a method that can estimate the performance on corpus B of an NLP tool trained on corpus A in an unsupervised way, i.e., without the necessity to annotate a part of B.

In related work, various properties of corpora have been used for making machine learners more adaptive (Ravi et al., 2008; Chen et al., 2009; McClosky, 2010; Plank & van Noord, 2010).

2. Experimental design

2.1. Corpus

We used part-of-speech data extracted from the British National Corpus (BNC, 2001) and consisting of written books and periodicals. The BNC annotators divided text from books and periodicals into 9 subcorpora. Since the BNC has been tagged automatically, the experiments in this article are artificial in the sense that they do not learn *real* part-of-speech tags but rather part-of-speech tags as they are assigned by the automatic taggers.

2.2. Part-of-speech taggers

The experiments carried out in the scope of this article are all part-of-speech (POS) tagging tasks. There are 91 different POS labels in the BNC corpus which are combinations of 57 basic labels. We used three algorithms to assign part-of-speech labels to the words from the test corpus: majority baseline, MBT (Memory-Based tagging) (Daelemans & van den Bosch, 2005), and SVMTool (Giménez & Márquez, 2004). The majority baseline algorithm assigns the POS label that occurs most frequently in the training set for a given word, to the word in the test set.

2.3. Similarity measures

To measure the difference between two corpora we implemented six similarity measures. For example, to calculate the Rényi divergence between corpus P and corpus Q the following formula is applied:

$$R\acute{e}nyi(P;Q;\alpha) = \frac{1}{(\alpha-1)} log_2 \left(\sum^k p_k^{1-\alpha} q_k^{\alpha} \right)$$

p_k is the relative frequency of a token k in the first corpus P, and q_k is the relative frequency of token k in the second corpus Q. α is a free parameter.

We tested 6 measures: Rényi, Variational (L1), Euclidean, Cosine, Kullback-Leibler, and the Bhattacharyya coefficient. For Rényi, we tested four different α-values: 0.95, 0.99, 1.05, and 1.1. We found that the Rényi divergence with $\alpha = 0.99$ resulted in the best linear relation between accuracy and divergence score according to the Pearson product-moment

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).

correlation. For majority this correlation was 0.91, for MBT 0.93, and for SVMTool 0.93.

2.4. Results and conclusions

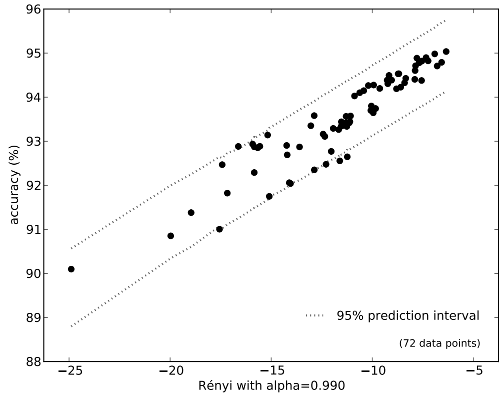


Figure 1. The varying cross-domain accuracy of the SVMTool POS tagger with varying distance between the training and test corpus.

For all pairs of domains, each of 5 subparts from the training domain is paired with each of 5 subparts from the testing domain. This results in a 25-fold cross-validation cross-domain experiment, which is depicted as a data point in Figure 1. The abscissa of a data point is the Rényi similarity score between the training and testing component of an experiment. The ordinate is the accuracy of the POS tagging experiment. The higher (less negative) the similarity score, the more similar training and testing data are.

The dotted lines are the 95% prediction intervals for every data point. These boundaries are obtained by linear regression using all other data points. The interpretation of the intervals is that any point, given all other data points from the graph, can be predicted with 95% certainty, to lie between the upper and lower interval boundary of the similarity score of that point. The average difference between the lower and the upper interval boundary is 4.36% for majority, 1.92% for MBT and 1.59% for SVMTool. This means that, when taking the middle of the interval as the expected accuracy, the maximum error is 0.8% for SVMTool. Since the difference between the best and worst accuracy score is 4.93%, using linear regression means that one can predict the accuracy three times better. We observed the same linear relation for in-domain experiments as for out-of-domain experiments. Giving an intuition about the sensitivity of the measure.

Our results show that it is feasible to find a linear correlation between the performance of POS taggers and the Rényi divergence and that the latter can therefore be used to predict the performance of the tagger for unseen out-of-domain data.

Acknowledgments

This research was made possible through financial support from the University of Antwerp (BIOGRAPH GOA-project).

References

BNC (2001). The British National Corpus, version 2 (BNC world). Distributed by Oxford University Computing Services on behalf of the BNC Consortium.

Chen, B., Lam, W., Tsang, I., & Wong, T.-L. (2009). Extracting discriminative concepts for domain adaptation in text mining. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 179–188). New York, NY, USA: ACM.

Daelemans, W., & van den Bosch, A. (2005). *Memory-based language processing.* Cambridge, UK: Cambridge University Press.

Giménez, J., & Márquez, L. (2004). SVMTool: A general POS tagger generator based on support vector machines. *Proceedings of the 4th International LREC* (pp. 43–46). Lisbon, Portugal: ELRA.

McClosky, D. (2010). *Any domain parsing: Automatic domain adaptation for natural language parsing.* Doctoral dissertation, Department of Computer Science, Brown University, Rhode Island, USA.

Plank, B., & van Noord, G. (2010). Dutch dependency parser performance across domains. *CLIN 2010: selected papers from the 20th CLIN meeting*, 123–138.

Ravi, S., Knight, K., & Soricut, R. (2008). Automatic prediction of parser accuracy. *Proceedings of the 2008 Conference on EMNLP* (pp. 887–896). Honolulu, Hawaii: Association for Computational Linguistics.

Diversity Measurement of Recommender Systems under Different User Choice Models

Keywords: recommendation, data mining, diversity measurement, simulation

Zoltán Szlávik Z.SZLAVIK@VU.NL
Wojtek Kowalczyk W.J.KOWALCZYK@VU.NL
Martijn Schut M.C.SCHUT@VU.NL
VU University Amsterdam, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands

1. Introduction

Recommender engines have made an overwhelming entrance into the world of digital information. For this, electronic commerce is a front-runner, hoping to sell more products using recommenders. Also, many owners of other information systems start to realise that recommenders may benefit the users in finding items of interest. Whether recommendations are generated by algorithms (Amazon, Netflix) or in more social way (e.g., Facebook, Twitter), owners ultimately want to know if their users can better and faster navigate through the available information. On commercial sites, this means whether users buy more items. Besides tackling technical questions, e.g., on deployment or algorithmic optimisation, owners of a recommender want to know how users react to their engine, and, following from that, how they or the engine can react to these users' reactions. This will enable them to dynamically adapt to the customer within the purchasing process – similar to how a human salesperson pitches in order to sell a product.

The objective of the study reported on in this paper [1] is to systematically investigate item-, user-, and rating-diversity effects of a movie recommender system by means of simulations. These simulations are based on real-world usage data (Bennett & Lanning, 2007). We aim to answer the following two questions: 1) *What happens if we force users to rate a certain number of items in a period of time (e.g., everyone rates 5 movies a week)?* Such a restriction is an example of how the owner of the recommender can 'influence' the behaviour of users. 2) *What is the effect of changing the type of information that a recommender gives to users?* For example, a recommender can show to the user either most popular movies, or movies that match best the user's preferences.

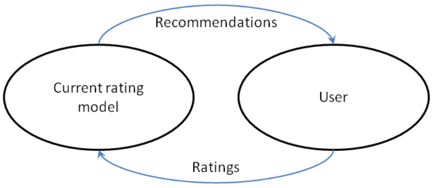


Figure 1. Movie- and user profiles, and movie selection and ratings affect each other.

2. Experiments

In order to study how a recommender system affects diversity, it is important to understand the circular nature of the 'life' of a recommender. On one hand, the recommendations a system provides to its users shape the way the users act (otherwise the system cannot be considered of good quality). On the other hand, the way users select and, in this case, rate items also affects the recommendations that the system gives in a later stage (assuming that the system is retrained regularly or adapts on-line). This circular process, illustrated in Figure 1, can be broken down into 'rounds'. First, the Recommender uses the Current Rating Model (CRM) to produce recommendations for a specific user. Then the User selects some of the recommended movies and rates them. The ratings are sent back to the Recommender and the CRM is updated. In this way, a user's behaviour affects the rating model and vice versa: the rating model has an impact on a user's ratings. To investigate how diversity changes over time, we ran simulations in which we controlled user behaviour. We did so by introducing several choice models that simulate

[1] Full paper: (Szlávik et al., 2011)

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).

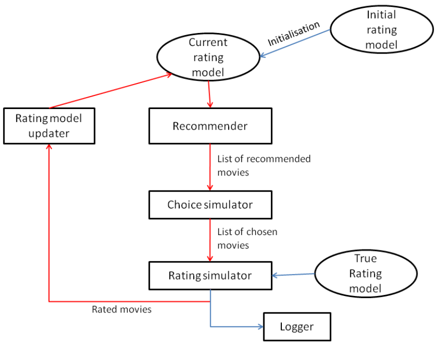


Figure 2. Simulation outline.

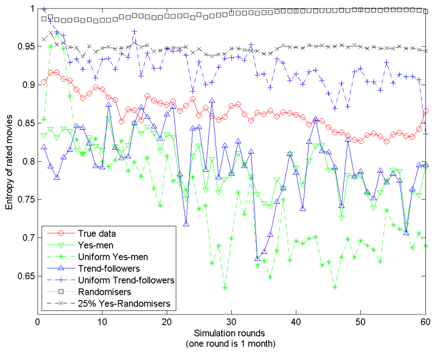


Figure 3. Entropy of rated movies.

how users make selections from a list of recommended items. We ran a number of simulated rounds one after another. The simulation output for each choice model was then analysed and compared, focusing on various senses of diversity. A round, shown as a loop in Figure 2, represents a time period of one month, i.e., we simulated recommendations and movie ratings of one month before updating the rating model and moving on to the next period. The recommender algorithm we used was based on the work by Funk (2006), simulations were aligned with the Netflix Prize data.

Choice models – We used six choice models, which resulted in six runs of simulation. We tested 3 basic choice models: *Yes-men* – each user of the recommender system completely accepts personalised recommendations; *Trend-followers* – everyone watches and rates movies that are highest rated on average; *Randomisers* – users are completely indifferent to recommendations. In addition, for the first two, we also looked at *uniform* variants, where the number of ratings per user follows a uniform distribution, i.e. every user rates the same number of movies. Finally, we looked at a mixture model (called *25% Yes-Randomisers*) where users accept the movies recommended to them, but only with a probability of 0.25.

Entropy – Among several measures reported in the full paper, we measured the *Normalised Shannon entropy of rated movies* in every round (Figure 3).

Results – The entropy of rated movies, shown in Figure 3, indicates that the distribution of movies when using the uniform versions of choice models is closer to being uniform themselves. This is particularly evident when we consider the differences between Trend-followers and Uniform Trend-followers, and not so much in case of the Yes-men models. When users can only choose from the same list of movies and they need to choose the same number of movies, the distribution of movies is even. However, if one of these uniformities is not true, i.e., they can either get personalised recommendations or choose as many movies as they want, entropy stays lower, meaning that there will be movies increasingly more/less popular over time.

3. Conclusions

We have investigated how diversity changes under different models of user behaviour when using a recommender system. Our simulations have practical implications, both for recommender system owners and designers. In particular, we have found that forced uniformity in terms of number of items rated does not necessarily result in users becoming more uniform, and the mean ratings they give to items will decrease, indicating lower system performance as perceived by the user. Also, we have identified how three kinds of choice models, i.e., Yes-men, Trend-followers and Randomisers, result in different diversity and mean rating values. This is a particularly important result as these behaviours can directly be encouraged by recommender system owners, e.g., we might decide to offer more trending items to one particular kind of users.

References

Bennett, J., & Lanning, S. (2007). The netflix prize. *In KDD Cup and Workshop in conjunction with KDD* (pp. 3–6).

Funk, S. (2006). Netflix Update: Try This at Home. `http://sifter.org/~simon/journal/20061211.html`.

Szlávik, Z., Kowalczyk, W., & Schut, M. (2011). Diversity measurement of recommender systems under different user choice models. *ICWSM.* Barcelona, Spain.

Customer Satisfaction and Network Experience in Mobile Telecommunications

Dejan Radosavljevik DRADOSAV@LIACS.NL

Peter van der Putten PUTTEN@LIACS.NL

Leiden Institute of Advanced Computer Science, Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands

Kim Kyllesbech Larsen KIM.LARSEN@T-MOBILE.NL

International Network Economics, Deutsche Telecom AG, Landgrabenweg 151, D-53227 Bonn, Germany

Keywords: Customer Satisfaction, Customer Experience, Network Experience, Social Networks

Extended Abstract

The purpose of this extended abstract is to serve as a problem statement for our intended research.

Mobile telecommunications services are increasingly becoming a commodity. In saturated markets with mobile telephone penetration above 100%, e.g. The Netherlands, new customers are hard to find. An operator, in order to grow, has to attract customers from competition, and at the same time retain its existing customer base.

Keeping the current customers satisfied can be a powerful means in achieving both these goals. Customer satisfaction is defined as a customer's overall evaluation of the performance of an offering to date (Johnson and Fornell 1991). But, just measuring customer satisfaction shows neither the ways of achieving it, nor ways of battling poor customer satisfaction. According to the proponents of Customer Experience Management, "customer satisfaction is essentially the culmination of a series of customer experiences or, one could say, the net result of the good ones minus the bad ones" (Meyer & Schwager, 2007). Customer experience is the internal and subjective response customers have to any direct or indirect contact with a company. Data about these experiences are collected at "touch points": instances of direct contact either with the product or service itself or with representations of it by the company or some third party (Meyer & Schwager, 2007). In our previous work, we have created a framework for measuring customer experience in Mobile Telecommunications (Radosavljevik, van der Putten & Kyllesbech Larsen, 2010), which we intend to use in this research as well.

For several reasons, the focus of the intended research will be on the Mobile Telecom network. First of all, the mobile network is the most frequent touch point between the customers and the operator. Most of the customers' experiences occur here. Furthermore, deteriorated network performance can be seen as a relational trigger for re-evaluation of the relationship with the operator (Gustafsson, Johnson, & Roos, 2005). For these reasons, the customers' network experience and their satisfaction with the network can be seen as drivers for the overall customer satisfaction with the operator.

Therefore, identifying network quality parameters that drive customer satisfaction or dissatisfaction, and their respective thresholds, is of high importance to operators, as they can serve as guidance for network improvements. We intend to achieve this in the following manner. First, we will perform a survey on a random customer sample, in order to establish their level of satisfaction with the mobile network. Next, the customers' network experience prior to the survey will be measured on the same sample and will focus mainly on the quality and quantity of interactions with the network, as well as the means for these interactions, the phone. Then, these values would be cross correlated to establish possible dependencies between customer satisfaction and network quality. Finally, we intend to build a predictive model using these network quality parameters, which in turn could be used to predict the satisfaction level of the entire customer base. Also, we intend to cluster customers based on network quality parameters and satisfaction.

We do not assume that all customer (dis)satisfaction with an operator's network stems from objective network quality parameters. First of all, as much as the customer's satisfaction with the operator's network can drive the overall customer's satisfaction, it is conceivable that an opposite causality may exist. An answer to this "chicken or egg" question could provide operators with guidelines for their investments efforts. Next, customers' peers can be a very influential factor as well. For example, some customers may have low satisfaction with the mobile telecom network, i.e. perception of its quality, not because of low values in their own network quality parameters, but because they are in the same social network with a person having a large number of network problems or with a person who has a perception of the telecom network quality as low. Contrary to this, a customer experiencing frequent network problems could be unaware of them, or ignore them, and still be satisfied with the network quality due

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands*.

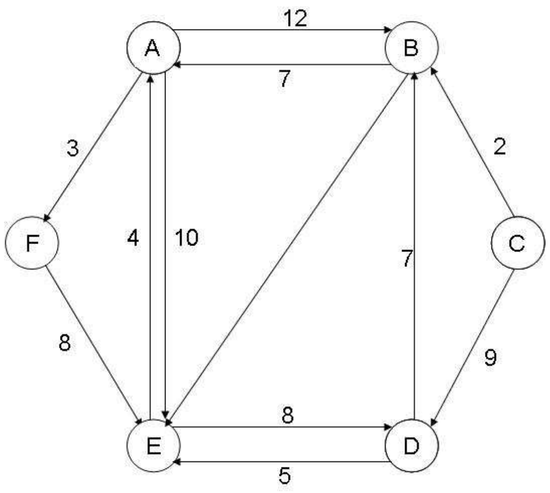


Figure 1. A Theoretical communication graph

to the influence of highly satisfied peers or because they are satisfied with the operator in general. For these reasons, the link between Customer Satisfaction and Network Experience in Telecommunications can also be formulated as a relational learning problem.

The relevance of the Social Networks within mobile telecom customers is already in the focus of research literature (Dasgupta, 2008). Therefore, we intend to derive a social network using the communication graph of the surveyed customers within a given timeframe, e.g. one month, and repeat the procedure we described in the previous paragraph, in order to establish how important is the social network effect on this phenomenon. A simplified version of this communication graph is shown on Figure 1. It is a directed weighted graph, where the direction depends on who initiated the communication (e.g. call, SMS). The graph can be weighted on different parameters such as the count of calls initiated by customer A to customer B, the duration of these calls, the number of SMS messages that customer A sent to customer B, or any combination of these parameters. Each of the nodes (customers) on this graph has attributes that are used in traditional data mining in telecommunications (Radosavljevik, van der Putten & Kyllesbech Larsen, 2010).

Finally, as future research, we intend to investigate the spreading of satisfaction and dissatisfaction through the graph, using diffusion theory as background. Here, we treat satisfaction and dissatisfaction separately because it is debatable whether dissatisfied customers engage in more Word-of-Mouth, and if so, by how much (Anderson, 1998)?

References

Anderson, E. (1998). Customer Satisfaction and Word-of-Mouth, *Journal of Service Research, 1 (1)*, 5–17.

Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukherjea, S., & Nanavati, A. A. (2008). Social Ties and their Relevance to Churn in Mobile Telecom Networks. *Proceedings of the 11th international conference on Extending database technology,* pp. 668-677.

Gustafsson, A., Johnson, M. D., Roos, I. (2005). The Effects of Customer Satisfaction, Relationship Commitment Dimensions, and Triggers on Customer Retention. *Journal of Marketing, 69*, 210–218.

Johnson, M. D. and Fornell, C. (1991). A Framework for Comparing Customer Satisfaction Across Individuals and Product Categories. *Journal of Economic Psychology, 12 (2)*, 267–286.

Meyer, C. & Schwager, A. (2007). Understanding Customer Experience. *Harvard Business Review, 85(2)*, 116-126.

Radosavljevik, D., van der Putten, P., Kyllesbech Larsen, K (2010). The Impact of Experimental Setup in Prepaid Churn Prediction for Mobile Telecommunications: What to Predict, for Whom and Does the Customer Experience Matter? *Transactions on Machine Learning and Data Mining, 3(2)*, 80-99.

Avoiding overfitting in surrogate modeling: an alternative approach

Keywords: Surrogate modeling, LRM, cross-validation, overfitting

Huu Minh Nguyen HUUMINH.NGUYEN@UGENT.BE
Ivo Couckuyt IVO.COUCKUYT@UGENT.BE
Yvan Saeys YVAN.SAEYS@UGENT.BE
Luc Knockaert LUC.KNOCKAERT@UGENT.BE
Tom Dhaene TOM.DHAENE@UGENT.BE
Ghent University - IBBT, Sint-Pietersnieuwstraat 25, 9000, Gent, Belgium

Dirk Gorissen DIRK.GORISSEN@SOTON.AC.UK
University of Southampton, Room 2041, Building 25, Highfield Campus, School of Engineering Sciences, University of Southampton, SO17 1BJ, UK

1. Introduction

In many simulation applications , performing routine design tasks such as visualization, design space exploration or sensitivity analysis quickly becomes impractical due to the (relatively) high cost of computing a single design (Forrester et al., 2008). Therefore, in a first design step, surrogate models are often used as replacements for the real simulator to speed up the design process (Queipo et al., 2005). Surrogate models are mathematical models which try to generalize the complex behavior of the system of interest, from a limited set of data samples to unseen data and this as accurately as possible. Examples of surrogate models are Artificial Neural Networks (ANN), Support Vector Machines (SVM), Kriging models and Radial Basis Function (RBF) models. Surrogate models are used in many types of applications, however in this work we concentrate on noiseless simulation data, as opposed to measurement data or data coming from stochastic simulators.

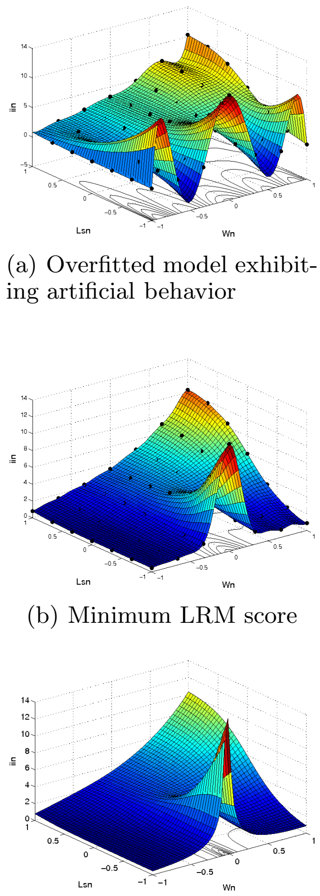
An important consideration when constructing the surrogate model is the selection of suitable hyperparameters, as they determine the behavior of the model. Finding a good set of hyperparameters is however nontrivial, as it requires estimating the generalization ability of the model. When dealing with sparse data, the search for good hyperparameters becomes even harder. Bad hyperparameters will lead to models with high training accuracy (as shown in Fig. 1(a)) but which fail to capture the true behavior (Fig. 1(c)) and instead exhibiting artificial ripples and bumps.

A common strategy for optimizing model hyperparameters when only a limited amount of training data is available, is cross-validation. However, because many models have to be trained, performing cross-validation can be quite time and resource consuming, especially if the cost of model building is high. Moreover, cross-validation is not always efficient at preventing artificial model behavior (Gorissen et al., 2009).

We present in this work a new generic auxiliary model selection measure, called the Linear Reference Model (LRM), which is designed to be fast to compute and which reduces the chance of overfitting. LRM identifies regions where the model exhibits complex behavior (such as oscillations) but lacks the data to support this and penalizes the model accordingly. Overfitted regions are identified by comparing the predicted output values of the surrogate model to that of a local linear interpolation. Large deviations are an indication of overfitting, and models are penalized more heavily if they diverge more from the local linear interpolation. Note that the risk of underfitting is usually neglible as the high accuracy typically required in surrogate modeling can only achieved by using high complexity models, in which case LRM will only reduce their tendency to overfit but never to the extent that the models will underfit. Fig. 1(b) shows the effect of applying the LRM measure. Although the training accuracy of the model is now worse, the intermediate surrogate model is better at capturing the true behavior of the system.

The LRM score is calculated as follows. First, a Delau-

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the authors(s)/owner(s).



(a) Overfitted model exhibiting artificial behavior

(b) Minimum LRM score

(c) True function

Figure 1. surrogate models of the input noise current ($\sqrt{i_{in}^2}$) of a Low Noise Amplifier (Gorissen et al., 2009) generated with different model selection criteria. The dots represent a sparse intermediate training during model construction (7×7 samples).

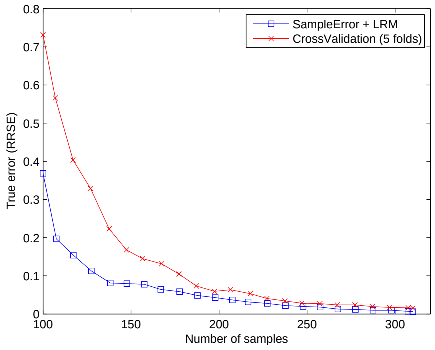


Figure 2. Academic example error on the independent test set as function of the number of data samples for "cross-validation" and "LRM in conjuction with the in-sample error".

nay tesselation of the input space is contructed after which, for each simplex given by the tesselation, a hyperplane through the corresponding samples is built. These hyperplanes form the local linear interpolation which will be used as reference for the surrogate model. Next, the surrogate model is compared with the local linear interpolation at every simplex, and the difference between the two is calculated. The LRM score for the surrogate model is then simply the average difference over all simplices.

We applied the LRM measure to both an analytic academic example and a real world application (Gorissen et al., 2009) using an adaptive model building scheme. In this scheme, a sequential sampling algorithm adds a small number of new samples to the training data at each iteration after which the surrogate model is rebuilt. Both cross-validation and LRM are then used to evaluate the updated models. Our experiments show that, in this context, the accuracies of the models selected by LRM converge faster and are better or comparable to accuracies of models selected using cross-validation. This is illustrated in Fig. 2 for the academic example, where the accuracy on an independent test set is plotted as a function of the number of selected samples. When the number of training samples is relatively small and the models are prone to overfitting, LRM (in combination with the in-sample error) achieves much lower errors than cross-validation. As the number of training samples increases, the difference in accuracy of the models selected by both approaches diminishes. However, LRM achieves this accuracy at much reduced computational cost and can thus provide an interesting alternative to cross-validation in simulation-based engineering design.

References

Forrester, A., Sobester, A., & Keane, A. (2008). *Engineering design via surrogate modelling: A practical guide.* Wiley.

Gorissen, D., De Tommasi, L., Crombecq, K., & Dhaene, T. (2009). Sequential modeling of a low noise amplifier with neural networks and active learning. *Neural Computing and Applications*, *18*, 485–494.

Queipo, N., Haftka, R., Shyy, W., Goel, T., Vaidyanathan, R., & Tucker, P. (2005). Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, *41*, 1–28.

Incorporating prior knowledge in multiple output regression with application to chemometrics

Jan Verwaeren JAN.VERWAEREN@UGENT.BE
Willem Waegeman WILLEM.WAEGEMAN@UGENT.BE
Bernard De Baets BERNARD.DEBAETS@UGENT.BE
KERMIT, Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Coupure links 653, Ghent, Belgium

Keywords: multiple output regression, chemometrics, compositional data, prior knowledge

Abstract

The incorporation of domain-specific knowledge can dramatically improve the predictive performance of any machine learning method. In this extended abstract, we discuss several types of prior knowledge that can be incorporated when handling multiple output regression problems, as well as a number of algorithms that are able to handle this prior knowledge. More precisely, we consider several sources of prior knowledge that can be incorporated when trying to derive the fatty acid composition of milk samples from their Raman spectra.

1. Introduction

It is generally known that the incorporation of domain knowledge in (existing) machine learning methods can improve their predictive performance. A popular way to incorporate prior knowledge consists in preprocessing the (input) data. As an example consider the well-studied problem of image classification. Typically, features that are relevant for the classification task at hand are extracted from the raw images and are subsequently used as input for a classification algorithm. Here the choice for a particular set of features is typically inspired by some sort of field knowledge. Data preprocessing, however, is not the only approach that can be used to incorporate prior knowledge into a machine learning method. Consider for example the multi-label classification setting. It has been shown (Hariharan et al., 2010) that introducing prior knowledge about label correlations can improve performance.

Multiple output regression has been studied extensively in the statistical literature. Several algorithms have been proposed, capable of incorporating or even implying different forms of field knowledge. Influenced by recent developments in the area of structured output prediction, existing approaches such as the maximum margin paradigm (Weston et al., 2005) and multiple output ridge regression (Cortes et al., 2007) have been the basis for the development of new multiple output regression algorithms, capable of incorporating field knowledge at different levels.

In this extended abstract, we consider a setting where the aim is to derive the fatty acid (FA) composition of a sample of milk, based on the Raman spectrum of that sample. More precisely, we want to estimate the relative proportions of each of 100 fatty acids, known to be present in cow milk. As input, the Raman spectrum of the fatty acid mixture, extracted from the milk sample, is used. As such, this task can be seen as a multiple output regression problem. Existing biological and chemical knowledge will be used to improve the performance of regression algorithms. It should be noted, however, that our discussion can be generalized to other regression problems as well.

2. Sources of field knowledge

When using the Raman spectrum to predict the fatty acid composition of milk, several sources of prior knowledge are available.

Functional data. The full Raman spectrum consists of measurements of the intensity of electromagnetic

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).

radiation at 3000 separate wavelengths, resulting in a high-dimensional input space. However, regarding spectral data as functional data allows for a significant reduction in the actual dimensionality of the data.

Chemical information. The molecular structure of all FAs potentially present in milk is known beforehand. Knowledge about the molecular structure of these FAs can be used to delineate 'important' regions within the spectrum for each FA.

Biological information. Most fatty acids that are present in cow milk are produced through some (often known) biological pathway. Moreover, most pathways are used to produce several FAs simultaneously, implying that relative proportions of these fatty acids are positively correlated. Enforcing this correlation during the learning procedure might thus improve the performance.

Compositional data. Since it is our aim to predict relative proportions, all proportions should add up to one, implying an *a priori* negative correlation between any pair of FAs. Here as well, taking these correlations into account can improve the performance (Verwaeren et al., 2011).

3. Incorporating knowledge using Joint Kernel Maps

Joint kernels provide an elegant way to embed knowledge on possible correlations between inputs, outputs or input-output pairs into a learning procedure (Tsochantaridis et al., 2005). Although initially introduced for discrete output spaces, joint kernels can be used in a multiple output regression setting as well. To illustrate the idea of the joint kernel map, consider a set of vector-valued input-output pairs $\{(\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_m,\mathbf{y}_m)\} \subset \mathbb{X}\times\mathbb{Y}$ (where $\mathbb{X}=\mathbb{R}^p$, $\mathbb{Y}=\mathbb{R}^q$). A feature mapping $\phi_{\mathbb{X},\mathbb{Y}}:\mathbb{X}\times\mathbb{Y}\to\mathbb{R}^N$ is called a joint feature map. The function $J:\mathbb{X}\times\mathbb{Y}\times\mathbb{X}\times\mathbb{Y}\to\mathbb{R}$ that computes the dot product in $\phi_{\mathbb{X},\mathbb{Y}}$ is called a joint kernel.

A multiple output regression algorithm, briefly described below, that can make use of this type of kernel was introduced by (Weston et al., 2005). Consider the linear map $\mathbf{y}(\mathbf{x})=W\mathbf{x}$ (where W is a $q\times p$ parameter matrix). The following large margin method can be used to estimate W for a given dataset:

$$\min_W \|W\|_{\mathrm{Fro}} \ ,$$

subject to

$$\|W\mathbf{x}_i-\mathbf{y}\|^2 \geq \|W\mathbf{x}_i-\mathbf{y}_i\|^2 \ , \qquad \forall i, \{\forall \mathbf{y}\in\mathbb{Y} : \|\mathbf{y}_i-\mathbf{y}\|\geq\epsilon\} \quad (1)$$

When restricting to the situation where all outputs are normalized ($\forall\mathbf{y}\in\mathbb{Y}:\|\mathbf{y}\|=1$), the dual form of (1) can be written using a joint kernel

$$\begin{aligned}\max_{\alpha_{i\mathbf{y}},\alpha_{j\hat{\mathbf{y}}}\geq 0} \frac{\epsilon^2}{4} & \sum_{i,\mathbf{y}:\|\mathbf{y}_i-\mathbf{y}\|\geq\epsilon} \alpha_{i\mathbf{y}} \\ -\frac{1}{2} & \sum_{\substack{i,\mathbf{y}:\|\mathbf{y}_i-\mathbf{y}\|\geq\epsilon\\ j,\hat{\mathbf{y}}:\|\mathbf{y}_j-\hat{\mathbf{y}}\|\geq\epsilon}} \alpha_{i\mathbf{y}}\alpha_{j\hat{\mathbf{y}}}\left[J((\mathbf{x}_i,\mathbf{y}_i),(\mathbf{x}_j,\mathbf{y}_j))\right. \\ & -J((\mathbf{x}_i,\mathbf{y}_i),(\mathbf{x}_j,\hat{\mathbf{y}})) - J((\mathbf{x}_i,\mathbf{y}),(\mathbf{x}_j,\mathbf{y}_j)) \\ & \left.+J((\mathbf{x}_i,\mathbf{y}),(\mathbf{x}_j,\hat{\mathbf{y}}))\right] \ . \quad (2)\end{aligned}$$

Optimization problems (1) and (2) are equivalent when $\phi_{\mathbb{X},\mathbb{Y}}$ is chosen as the tensor product between $\mathbf{x}$ and $\mathbf{y}$. Known correlations between inputs and outputs can be incorporated by choosing an appropriate type of kernel (e.g. the patchwise kernels presented in (Weston et al., 2005)).

4. On the poster ...

In this extended abstract, we advocated the idea of incorporating different types of prior knowledge in a learning procedure. In a typical application of the multiple output regression setting in chemometrics, different sources of prior knowledge were reviewed and a maximum margin algorithm that is able of incorporating this prior knowledge was described. On our poster we aim to present some (preliminary) results, demonstrating the usefulness of such an approach.

References

Cortes, C., Mohri, M., & Weston, J. (2007). *Predicting structured data*, chapter A General Regression Framework for Learning String-to-String Mappings, 143–168. MIT Press.

Hariharan, B., Zelnik-Manor, L., Vishwanathan, S., & Varma, M. (2010). Large scale max-margin multi-label classification with priors. *Proceedings of the International Conference on Machine Learning.*

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.

Verwaeren, J., Waegeman, W., & De Baets, B. (2011). Learning partial ordinal class memberships with kernel-based proportional odds models. *Computational statistics and data analysis*, *doi:10.1016/j.csda.2010.12.007.*

Weston, J., Schölkopf, B., & Bousquet, O. (2005). Joint kernel maps. *Proceedings of the 8th International Conference on Artificial Neural Networks, LNCS 3512.*

Detection of Developmental Stage in Zebrafish Embryos in a High throughput Processing Environment

Keywords: Neural Networks, Segmentation, Feature Extraction, High Throughput

Alexander Nezhinsky ANEZHINS@LIACS.NL
Irene Martorelli IMARTORE@LIACS.NL
Fons Verbeek FVERBEEK@LIACS.NL
Universiteit Leiden, Leiden Institute of Advanced Computer Science, Niels Bohrweg 1, 2333CA Leiden, the Netherlands

Abstract

The transparency and fast development of zebrafish embryo make it a valuable choice for studying its morphology during growth. For a high throughput approach of the analysis of zebrafish embryos there is a need for automated detection of the developmental stage. So, given an input image containing a zebrafish embryo the analysis should produce an accurate estimation of its developmental stage. To this end we have designed an algorithm that can potentially accomplish this. Preliminary experimental results are promising.

1. Introduction

The screening of Danio rerio (zebrafish) is used in many high throughput (HT) applications for various studies, including compound screening and target prediction. However, determination of the developmental stage of each embryo in a HT setting is hampered, because a human expert is needed to identify the stage correctly. In this paper we introduce a framework for automatic screening and analysis of zebrafish embryos. This includes both segmentation and learning algorithms with a development stage detection capability. We exploit the fact that the developmental stage of a zebrafish embryo can be identified by its shape (Kimmel *et al.*, 1995).

2. Methods

2.1 Zebrafish embryos

We have analyzed Danio rerio embryos that were between the 8cell (about 1.25h past fertilization) and the 30% epiboly stage (about 5 hpf). The embryos were positioned with their animal pole approximately on top and their vegetal pole on bottom in order to get them in a close to lateral view.

2.2 Image acquisition

The images were acquired using Leica MZ16FA light microscope in 24-bit color at a resolution of 2592 × 1944 pixels. Time–lapse acquisition is accomplished in automated fashion over time $T(1 – 9.5h)$, Δt=5min. Each image contains one embryo shape located at a random position in the plane.

3. Approach

The proposed framework consists of the following steps (Figure 1). First step is to segment the shape of a zebrafish embryo from input images and remove the background despite position, rotation and scale.

In the next step feature extraction is applied to the localized shape.

The features that are obtained in the first step will serve as input to a back propagation Neural Network (NN).

The developmental stage is known in the training phase and it is used to determine an error value. After the learning phase the NN can be used for developmental stage recognition.

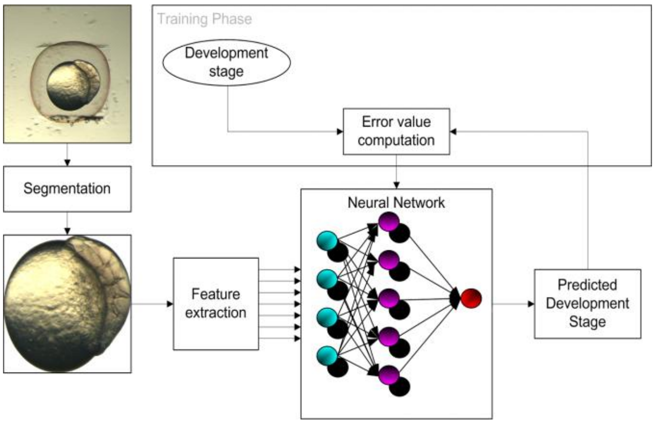


Figure 1. Overview of the proposed framework for automatic zebrafish embryo developmental stage determination.

3.1 Segmentation

Each zebra fish embryo within an image is surrounded by a membrane. Therefore, it can be seen as a close to

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the author(s)/owner(s).

circular convex shape located within another convex shape. As we are interested only in the embryo itself we need to remove both the background and the membrane. We have used a threshold method, based on the Otsu's (Otsu *et al*, 1978) algorithm and applied it to grayscale images to extract the embryo shape.

3.2 Feature extraction

After the zebrafish embryo shape is extracted from the input image we transfer this graphical representation into feature space. Since every embryo is rotated differently and the rotation angle is not known beforehand we specifically focus on rotation invariant shape features (Putten *et al.*, 2007). To incorporate rotation independent texture and color features we use the color ring projection (Tsai *et al.*, 2002). An infogain analysis is performed to sort out the strongest features.

3.3 Machine Learning

Neural Networks (NNs) have been proven to be successful at classification tasks where large numbers of parameters are involved and are stable to noise, especially in image recognition cases (Koval *et al.*, 2003). We use a back propagation NN (Mitchell *et al*, 1997) with 80 hidden nodes.

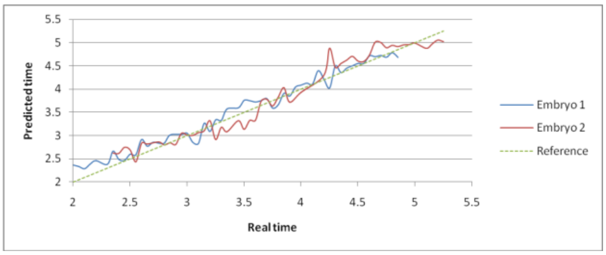
For the training phase we use embryos of which the developmental stage is known a priori. We feed the extracted features together with the developmental stage as parameters to the NN. For the test phase we let the NN calculate the developmental stage based on the input parameters.

4. Results and future work

For the segmentation step we have evaluated different threshold based methods based on grayscale images, RGB and HSI color space (Gonzales *et al.*, 2001) and edge maps (Canny *et al.*, 1986). We also evaluated shape based methods like the circular Hough transform (Shapiro *et al.*, 2001) and our convex shape retrieval method (Nezhinsky *et al.*, 2010). Threshold based method with Otsu's threshold function (Otsu, 1978) proved to be most effective for our dataset (about 76% of the images were segmented correctly) the rest was discarded in an automated fashion by analyzing the segmented shape size.

We have used the basic features as described in **3.2** as input vector during the training of a NN. We have chosen to use a NN as initial classification technique, as our case, the features have not been studied in more detail yet.

The preliminary results are shown in Plot 1. The plot gives the insight of the real embryo age against the NN predicted age. As can be seen the results are promising, yet the training set is too small (41 embryos giving 6336 data instances) and they have to be compared to other data mining methods in more detail, more input data must be generated and a more detailed feature analysis and feature selection should be done.



Plot 1. Age prediction from 2 to 5.5hpf for 2 embryos.

Acknowledgments

This work was partially supported by the Smartmix Programme and the European Union Mobility fund.

References

Canny, J. (1986). *A computational approach to edge detection.* IEEE Trans. Pattern Analysis and Machine Intelligence, vol 8, pages 679-714

Gonzales, R., Woods, R. (2001). *Digital Image Processing.* Addison-Wesley, London, 2nd edition.

Kimmel, C. B., Ballard, W. W., Kimmel, Ullman, B., Schilling, T. F. (1995). *Stages of Embryonic Development of the Zebrafish.* Developmental Dynamics,203:253–310.doi: 10.1002/aja.1002030302

Koval, V., Turchenko, V., Kochan, V., Sachenko, A., Markowsky, G. (2003). *Smart License Plate Recognition System Based on Image Processing Using Neural Network.* IEEE Int. Worksh on Intel. Data Acq. and Adv. Comp. Sys., Lviv, UA

Mitchell, T. (1997). *Machine Learning*, McGraw Hill. ISBN 0-07-042807-7.

Nezhinsky, A. E., Kruisselbrink, J., Verbeek, F.J. (2010). *Convex Shape Retrieval from Edge Maps by the use of an Evolutionary Algorithm.* 1st International Conf. on Bioinf., Valencia, Spain Proc.Bioinf., (Fred,A., Filipe,J., Gamboa,H.): 221-225

Otsu, N. (1978). *A threshold selection method from gray-level histogram*, IEEE Transactions on Systems Man Cybernet.

Putten, P.W.H. van der , Bertens, L.M.F. , Liu, J. , Hagen, F , Boekhout, T. & Verbeek, F.J. (2007), *Classification of Yeast Cells from Image Features to Evaluate Pathogen Conditions.* In Hanjalic, A., Schettini, R., Sebe, N. (Ed.), SPIE Vol. 6506.

Shapiro, L., Stockman, G. (2001). *Computer Vision.* Prentice-Hall, Inc. 2001

Tsai D.-M., Tsai Y.-H. (2002). *Rotation-invariant pattern matching with color ring-projection Pattern Recognition,* 35 (1), pp. 131-141.

Zhai, L., Dong, S., and Ma, H. (2008). *Recent Methods and Applications on Image Edge Detection.* Proc. of the 2008 intl. Workshop on Education Technology and Training & 2008 intl. Workshop on Geoscience and Remote Sensing-Vo01. ETTANDGRS. IEEE Computer Society, Washington, DC, 332-335.

Fractionally Predictive Spiking Neural Networks

Sander M. Bohte CENTRUM WISKUNDE & INFORMATICA, AMSTERDAM

Keywords: spiking neural networks, neural coding, error-backpropagation, signal processing

Extended Abstract[1]

In the late 1990's, findings from experimental and theoretical neuroscience suggested that the precise timing of individual spikes as emitted by real, biological neurons can be important in their information processing. This notion of neural spike-time coding has resulted in the development of a number of spiking neural network approaches demonstrating that spiking neurons can compute using precisely times spikes similar to traditional neurons in neural networks (Natschläger & Ruf, 1998; Bohte et al., 2002). However, in spite of the successes of spiking neural networks, and the theoretical appeal of spike-time coding, it has remained a challenge to extend spike-based coding to computations involving longer timescales without sacrificing any notions of information being coded in the timing of individual spikes.

Based on recent neuroscience findings (Lundstrom et al., 2008), and reconciling the notions of both spike-time coding and spike-rate coding, a novel scheme for spike-based neural coding is proposed based on the observation that a mild derivative – a *fractional* derivative – of a signal can under certain conditions *be* a series of spikes (Bohte & Rombouts, 2010). In this framework, neural spiking is a statistically efficient means of encoding time-continuous signals. It does so by approximating the internally computed neural signal as sum of shifted kernels, where these kernels decay following a power-law (e.g. figure 1A). Power-law kernels provide much longer traces of past signals as compared to exponentially decaying kernels, and are thus much more suitable for computing temporal functions over behaviorally relevant timescales. We demonstrate this by learning functions through time in a network of fractionally predictive spiking neurons, using error-backpropagation.

Fractionally predictive spiking neurons. A fractionally predictive spiking neuron j approximates the internal signal $y_j(t)$ with $\hat{y}_j(t)$ as a sum of shifted power-law kernels centered at spike-times $\{t_i\}$:

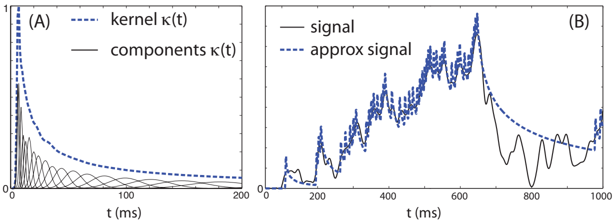


Figure 1. (A) Power-law kernel (dashed blue) for $\beta = 0.8$, and Guassian components κ_k (B) Signal (black) approxiamated with a sum of power-law kernels (dashed blue).

$$y_j(t) \approx \hat{y}_j(t) = \sum_{t_j<t} \kappa(t - t_j).$$

The fractional derivative of order α of this approximation $\hat{y}_j(t)$ is just the spike-train $\{t_i\}$ when the kernel $\kappa(t)$ decays proportional to a power-law $\kappa(t) \propto t^{-\beta}$ when $\alpha = 1 - \beta$ (Bohte & Rombouts, 2010):

$$\frac{\partial^\alpha \hat{y}_j(t)}{\partial t^\alpha} = \sum_{t_j<t} \delta(t - t_j).$$

Such signal approximation $\hat{y}_j(t)$ can be achieved by in a spiking neuron by computing the difference between the current signal estimation and the (emitted) future estimation (prediction) $\hat{y}(t)$, adding a spike t_i when this difference exceeds a threshold ϑ.

With a single, positive threshold only positive signal deviations are transmitted, and for negative deviations the transmitted signal decays as $t^{-\beta}$ (closely matching actual spiking neuron behavior (Pozzorini et al., 2010)). Such signal approximation is shown in figure 1B. For a variation on this schema, fractional spiking neurons require about half the spikes to encode a (self-similar) signal at the same signal-to-noise ratio as spiking neurons using exponential kernels (Bohte & Rombouts, 2010). The notion of fractional spiking neurons furthermore offers a straightforward way for neurons to carry out spectral filtering, and the code corresponds closely to biologically observed effects of spike-rate-adaptation (Bohte & Rombouts, 2010).

[1] Appearing in Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands. Copyright 2011 by the author(s)/owner(s). This is an abstract of (Bohte & Rombouts, 2010) and (Bohte, 2011)

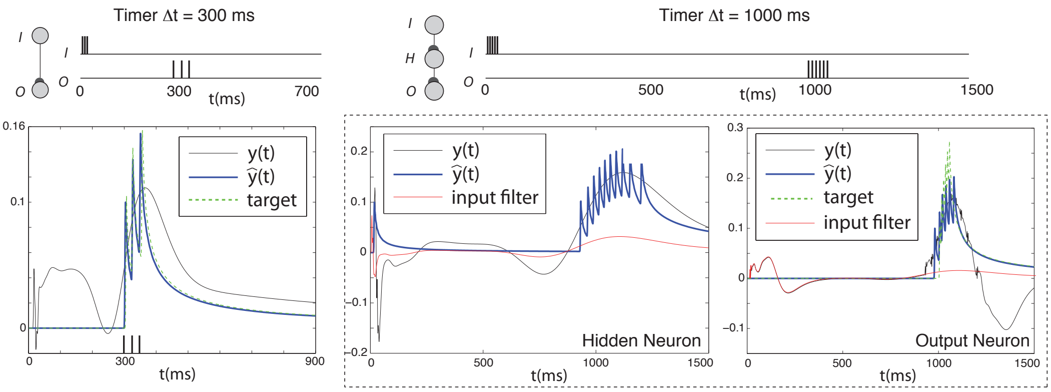


Figure 2. Neural timers. Left: learning a 300ms delay directly from input. Top: input-target spike pattern. Bottom: Learned response after 400 epochs, with the kernel approximation $\hat{y}(t)$ matching the target output. Target (green dashed) and internal signal $y(t)$ (solid black) and signal kernel approximation $\hat{y}(t)$ (dashed blue). Right: learning a larger response at $\Delta t = 1000$ms, with an additional hidden neuron. Top: input-target spike pattern. Bottom: Learned response after 900 epochs, for the hidden neuron (center) and the output neuron (right). Shown in red is the developed temporal filter.

Error-backpropagation Key properties of the fractional-spike coding framework can be exploited (Bohte, 2011) to learn functions over behaviorally relevant timescales. We derive error-backpropagation in the fractional spike-coding paradigm, and show that it allows spiking neural networks to learn functions through time like delayed timer-functions and recall tasks like delayed-match-to-sample.

Following standard neural networks notation, an artificial neuron j computes an internal variable $y_j(t)$ as a function over the weighted sum of filtered inputs $x_j(t)$: $y_j(t) = \mathcal{F}(x_j(t))$ and $x_j(t) = \sum_{i \in \mathcal{J}} w_{ij} f_i(y_i(t))$, where $\mathcal{J}$ is the set of presynaptic neurons i to neuron j, and $f_i(y_i(t))$ denotes the (temporal) input filter.

To implement a continuous-time ANN with fractionally predictive spiking neurons, we use the fact that a power-law kernel $\kappa(t)$ can be approximated as a sum (or cascade) of different, weighted exponentially decaying functions $\kappa_k(t)$: $\kappa(t) \approx \sum_k \kappa_k(t)$, as illustrated in figure 1A. This lets us rewrite $\hat{y}_i(t)$ as a sum of components $y_i^k(t)$:

$$\hat{y}_i(t) = \sum_{t_i < t} \sum_k \kappa_k(t - t_i) = \sum_k y_i^k(t).$$

At a receiving neuron j, a temporal filter κ_{ij} of the signal $\hat{y}_j(t)$ from neuron j can then be created by weighing these components with weights w_{ij}^k at the receiving synapse:

$$\kappa_{ij}(t) = \sum_k w_{ij}^k y_i^k(t).$$

and the neuron's input is thus computed as:

$$x_j(t) = \sum_{i \in \mathcal{J}} \sum_{t_i < t} \kappa_{ij}(t - t_i) = \sum_{i \in \mathcal{J}} w_{ij}^k y_i^k(t).$$

(for $w_{ij}^k = w_{ij} \forall\, k$, input neuron j decodes a weighted version of the output of presynaptic neuron i).

We use a standard fully connected feedforward neural network, with layers $\mathcal{I}$, $\mathcal{H}$, and O, populated with neurons i, j and m. Error-backpropagation adjusts the components w_{ij}^k of the filtering kernels $\kappa_{ij}(t)$ for each connection in the network: given desired output activation pattern $s_k(t)$ for each output neuron k, we define a standard quadratic error measure in terms of the output $\hat{y}$:

$$E(t) = \sum_{m \in \mathcal{O}} \left(s_m(t) - \hat{y}_m(t)\right)^2$$

The goal is to adjust each weight w_{ij}^k (and w_{jm}^k) in the network so as to minimize the error over some time-period $[T, T']$; given the defined correspondences, the actual gradient is straightforward (Bohte, 2011). Figure 2 shows an example where the learning rule is applied to learning a delayed timer function.

References

Bohte, S. (2011). Error-backpropagation in multi-layer networks of spiking neurons. *ICANN 20* (to appear).

Bohte, S., Kok, J., & La Poutre, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, *48*, 17–37.

Bohte, S., & Rombouts, J. (2010). Fractionally predictive spiking neurons. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel and A. Culotta (Eds.), *Nips 23*, 253–261.

Lundstrom, B., Higgs, M., Spain, W., & Fairhall, A. (2008). Fractional differentiation by neocortical pyramidal neurons. *Nature neuroscience*, *11*, 1335–1342.

Natschläger, T., & Ruf, B. (1998). Spatial and temporal pattern analysis via spiking neurons. *Network: Computation in Neural Systems*, *9*, 319–332.

Pozzorini, C., Naud, R., Mensi, S., & Gerstner, W. (2010). Multiple timescales of adaptation in single neuron models. *Front. Comput. Neurosci. Conference Abstract: Bernstein Conference on Computational Neuroscience.*

Granularity based Instance Selection

Nele Verbiest NELE.VERBIEST@UGENT.BE
Chris Cornelis CHRIS.CORNELIS@UGENT.BE
Department of Applied Mathematics and Informatics, Ghent University, 9000 Gent, Belgium

Francisco Herrera HERRERA@DECSAI.UGR.ES
Department of Computer Science and Artificial Intelligence, University of Granada, E-18071 Granada, Spain

Keywords: instance selection, fuzzy rough sets, granularity, nearest neighbour classification

Abstract

As more and more high-dimensional data are becoming available, interest rises in techniques that can remove noisy instances from them. We have designed an instance selection algorithm that is based on fuzzy rough sets and takes into account the granularity of the similarity measure used, and evaluate its impact on nearest neighbour classification.

1. Introduction

Rough sets (Pawlak, 1982) have been used widely to reduce the number of attributes (attribute selection (Bazan et al., 2000)) in datasets. Extending rough sets to fuzzy rough sets (Dubois & Prade, 1990) and using them for attribute selection has been explored extensively (Cornelis et al., 2010), but using fuzzy rough sets for instance selection is still at its infancy. In (Jensen & Cornelis, 2010), Fuzzy Rough Instance Selection (FRIS) was proposed. In this work, we extend this instance selection algorithm to a wrapper that takes into account the granularity of the similarity relation used.

2. Preliminaries: Fuzzy Positive Region, FRIS

We consider a decision system $(U, A \cup \{d\})$ that consists of a universe of instances $U = \{x_1, \ldots, x_m\}$ and a set of attributes $A = \{a_1, \ldots, a_n\}$ together with a fixed attribute d, called the decision attribute. An instance x in the universe has a value $a(x)$ for each attribute a in A that can be real or nominal. We assume that $d(x)$, the class of x, is always nominal.

In case A only contains nominal attributes, both A and d impose an equivalence relation on the universe, denoted by R_A and R_d. The equivalence classes of an instance $x \in U$ are denoted as $[x]_A$ and $[x]_d$. In order to approximate $[x]_d$, we can use its so-called R_A-lower approximation:

$$R_A \downarrow [x]_d = \{y \in U | [y]_A \subseteq [x]_d\}.$$

The positive region of the decision system containing all instances for which R_A can decide the decision class unequivocally is then given by:

$$POS_A = \bigcup_{x \in U} R_A \downarrow [x]_d.$$

In case A also contains continuous attributes, it is better to use a fuzzy similarity relation $R_A^\alpha : U \times U \to [0,1]$ that expresses for each pair of instances x and y the extent $R_A^\alpha(x,y)$ to which they are related to each other with respect to A. The granularity parameter α expresses how much two objects need to differ in order to discern them.

Using this fuzzy relation and a fuzzy implication[1] I, we can define the fuzzy lower approximation of $[x]_d$ (Radzikowska & Kerre, 2002):

$$(R_A^\alpha \downarrow [x]_d)(y) = \inf_{z \in U} I(R_A^\alpha(y,z), [x]_d(z)),$$

where $[x]_d(z)$ is 1 if $z \in [x]_d$ and 0 otherwise. The positive region can finally be defined as:

$$POS_A^\alpha(y) = \sup_{x \in U} (R_A^\alpha \downarrow [x]_d)(y),$$

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the author(s)/owner(s).

[1] A fuzzy implication is a mapping $I : [0,1]^2 \to [0,1]$ that extends the boolean implication. For instance, the Lukasiewicz implication is defined by: $I_L(x,y) = \max(1, 1 - x + y)$.

and expresses for each instance the extent to which it belongs to the (fuzzy) positive region.

The FRIS algorithm is based on the notion of fuzzy positive region: it removes all instances x for which $POS_A^{\alpha}(x)$ is smaller than 1. The idea is that instances not fully belonging to the positive region create inconsistencies and should be removed.

3. FRIS wrapper

It is clear that FRIS depends on the choice of the granularity α in the definition of the fuzzy positive region. When the granularity is smaller, R_A^{α} returns higher values, which means that the positive region is lower and more instances are removed. We observed experimentally that the optimal granularity depends on the dataset used. Therefore, we designed an algorithm that determines the granularity α based on the training data of the dataset. The pseudocode is given in Algorithm 1. $FRIS(\alpha)$ returns the instances selected by the FRIS procedure using α for the similarity measure. Thus, for each α in the given range, the FRIS wrapper performs instance selection using FRIS. The granularity for which the training accuracy is optimal is chosen. The training accuracy is determined by classifying the entire training dataset using the 1NN classifier in the reduced training dataset and calculating the proportion of correctly classified objects. Finally, FRIS is performed using this optimal granularity.

In order to evaluate the performance of the FRIS wrapper we used it to reduce 39 benchmark classification datasets from the UCI data repository. We used a 10 fold cross validation procedure and classified the test data using the 1NN classifier on the reduced training data. The test accuracy is calculated as the percentage of correctly classified test instances. The range of alphas used is different for each dataset but we do not go into detail on how to determine it in this work.

We compared our algorithm with 42 state of the art instance selection algorithms (e.g. DROP, CNN, CHC, ...). The non-parametrical Friedman test (Garcia et al., 2010) finds statistical significant differences between the algorithms. The FRIS wrapper has the best average ranking (8.18) among all algorithms. We also compared the FRIS wrapper to each of the other algorithms using the Wilcoxon test (Wilcoxon, 1945). The FRIS wrapper outperforms 39 out of the 42 algorithms at the 10% significance level. The other 3 algorithms are not significantly worse or better than our algorithm.

Currently, we are improving the algorithm by combining it with condensation methods (e.g. (Hart, 1968)), studying the range of alphas to be used and optimizing the running time.

Algorithm 1 FRIS wrapper

```
Input: $(U, A \cup \{d\})$, a range of alphas $\{\alpha_1, \ldots, \alpha_k\}$
accuracy.optimal = 0
$\alpha$.optimal = $\alpha_1$
for $\alpha = \alpha_1$ to $\alpha_k$ do
  if trainaccuracy(FRIS($\alpha$))>accuracy.optimal then
    accuracy.optimal = trainaccuracy(FRIS($\alpha$))
    $\alpha$.optimal = $\alpha$
  end if
end for
return FRIS($\alpha$.optimal)
```

Acknowledgments

Chris Cornelis would like to thank the Research Foundation Flanders for funding his research.

References

Bazan, J., Nguyen, H., Nguyen, S., Synak, P., & Wroblewski, J. (2000). *Rough set algorithms in classification problem*, 49–88. Heidelberg, Germany, Germany: Physica-Verlag GmbH.

Cornelis, C., Jensen, R., Hurtado, G., & Slezak, D. (2010). Attribute selection with fuzzy decision reducts. *Information Sciences*, *180*, 209 – 224.

Dubois, D., & Prade, H. (1990). Rough fuzzy sets fuzzy rough sets. *International Journal of General Systems*, *17*, 191,209.

Garcia, S., Fernandez, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence data mining: Experimental analysis of power. *Inf. Sci.*, *180*, 2044–2064.

Hart, P. E. (1968). The condensed nearest neighbour rule. *IEEE Transactions on Information Theory*, *18*, 515–516.

Jensen, R., & Cornelis, C. (2010). Fuzzy-rough instance selection. *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on Computational Intelligence* (pp. 1–7).

Pawlak, Z. (1982). Rough sets. *International Journal of Computer Information Science*, *11*, 341–356.

Radzikowska, A., & Kerre, E. (2002). A comparative study of fuzzy rough sets. *Fuzzy Sets Systems*, *126*, 137 – 155.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*, 80–83.

A zealous parallel gradient descent algorithm

Gilles Louppe and Pierre Geurts {G.LOUPPE, P.GEURTS}@ULG.AC.BE
Department of EE and CS & GIGA-R, University of Liège, Sart Tilman B28, B-4000 Liège, Belgium

Abstract

Parallel and distributed algorithms have become a necessity in modern machine learning tasks. In this work, we propose a zealous parallel stochastic gradient algorithm that minimizes the idle time of processors to achieve a substantial speedup.

1. Motivation

Massive datasets have become ubiquitous in many domains and traditional machine learning techniques are no longer suited to handle them properly. In practice, online and mini-batch gradient descent algorithms already allow one to tackle any huge dataset. However, those algorithms are inherently serial and hence are still prone to prohibitive computing times. In this work, we try to overcome this problem by parallelizing the computation across several processors in the context of shared memory architectures.

2. Zealous parallel gradient descent

In many machine learning algorithms, the task of training a model often reduces to an optimization problem whose objective is to minimize $\mathbb{E}_z[C(\theta, z)]$ where C is some (typically convex) cost function and the expectation is computed over training points z. In mini-batch gradient descent, θ is iteratively updated after a small number b of training examples, using $\theta_{k+1} := \theta_k - \alpha \sum_{t=s_k}^{s_k+b} \frac{\partial C(\theta_k, z_t)}{\partial \theta}$ as update rule.

In a shared-memory environment, mini-batch gradient descent can be parallelized as proposed in (Nedic et al., 2001; Gimpel et al., 2010; Louppe, 2010). In this setting, θ is stored in shared memory and mini-batches are processed asynchronously and independently by multiple processors. Once a processor finishes its current mini-batch, it updates θ in mutual exclusion using a synchronization lock and then proceeds to the next mini-batch until some convergence criterion is met. As it can be noted, this algorithm do not simulate the same mini-batch procedure that would happen on a single processor. In particular, some delay might occur between the time gradient components are computed and the time they are eventually used to update θ. This amounts to say that processors might use slightly stale parameters that do not take into account the very last updates. Yet, Nedic et al. (2001) and Zinkevich et al. (2009) showed that convergence is still guaranteed under some conditions. This asynchronous algorithm works well but suffers from a bottleneck which impairs the gains due to the parallelization. Contention might indeed appear on the synchronization lock, hence causing the processors to queue and idle.

To solve this problem, we propose the following more resource-efficient scheme. First, instead of blocking on the synchronization lock if some other processor is already in the critical section, we make processors skip the update altogether. In that case, a processor directly and zealously proceeds to the next mini-batch and locally queues its updates until it eventually enters the critical section. Second, the access policy to the critical section is changed so that access is granted only to the processor with the most queued updates. The combination of these two strategies prevents processors from idling and limits at the same time the effects of increasing the delay between updates of θ. The procedure is summarized below.

```
// Thread procedure
shared $\theta$;
$\Delta\theta \leftarrow 0$;
while not converged($\theta$) do
    mini-batch $\leftarrow$ get next mini-batch;
    for all z $\in$ mini-batch do
        $\Delta\theta \leftarrow \Delta\theta + \frac{\partial C(\theta,z)}{\partial\theta}$;
    end for
    if trylock(pid) then
        $\theta \leftarrow \theta - \alpha\Delta\theta$;
        $\Delta\theta \leftarrow 0$;
        next(pid);
    end if
end while
```

```
// Access policy
shared next $\leftarrow$ 0;
shared counter $\leftarrow$ new int[np];
function trylock(pid)
    counter[pid]++;
    return next == pid;
end function
function next(pid)
    counter[pid] $\leftarrow$ 0;
    next $\leftarrow$ arg max(counter);
end function
```

Appearing in *Proceedings of the 20^{th} Machine Learning conference of Belgium and The Netherlands.*

3. Results

The experiments presented below measure the speedup obtainable by our zealous parallel gradient descent algorithm in comparison with the speedup achieved by the asynchronous algorithm of (Nedic et al., 2001; Gimpel et al., 2010; Louppe, 2010). The effects of delaying the updates of θ on the convergence of the method were also observed and evaluated.

We chose to evaluate our algorithm in the context of training a (conditional) restricted Boltzmann machine on a large collaborative filtering task, as introduced in (Louppe, 2010; Salakhutdinov et al., 2007). In that typical setting, θ usually counts millions of values, which makes updates actually fairly expensive in terms of computing time and hence increases the likelihood of blocking on the synchronization lock.

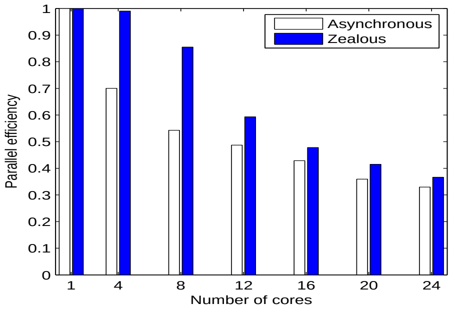


Figure 1. Parallel efficiency of the zealous algorithm.

Figure 1 illustrates the parallel efficiency of both asynchronous and zealous algorithms. The parallel efficiency is the speedup divided by the number of processors. It measures how well the parallel algorithms benefit from extra processors. We first observe that the zealous algorithm does indeed execute faster than the asynchronous algorithm. With 4 cores, the parallel efficiency of the zealous algorithm is nearly optimal, and then remains as expected significantly higher than the efficiency of the asynchronous algorithm as the number of cores increases. We also find that the gains in terms of speedup decrease as the number of cores increases. This is actually not surprising and can be explained using Amdahl's argument: the speedup of a parallel algorithm is bounded by the portion of code which is inherently serial. In our case, updates of θ remain serial, which explains the limit observed.

Figure 2 illustrates the error curve of the model on an independent dataset (i.e., the probe set of the Netflix dataset) when trained with an increasing number of cores. It highlights the fact that the zealous algorithm does indeed converge faster than the asynchronous algorithm in terms of wall clock time. For instance, we observe that the zealous algorithm run with 4 and 8 cores converges nearly as fast as the asynchronous algorithm run with 8 and 16 cores. For 20 and 24 cores however, significant oscillations can be observed in the learning curves of the zealous algorithm. Actually, this is a manifestation of the effects of delaying the updates of θ. Indeed, the more cores there are, the more likely they will skip the update of θ and hence the more they will accumulate gradient components. This accumulation of stale gradients causes inertia towards old directions and becomes more and more harmful as the algorithm approaches to the optimum. This explains the oscillations and also why they appear only after some time.

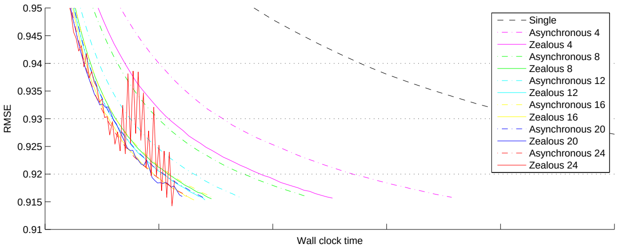


Figure 2. Learning curves of the zealous algorithm.

4. Future work

We would like to explore strategies to counter the effect of the delay and corroborate the results of this work on other machine learning tasks. We also plan to investigate other distributed architectures in continuation to our work in the context of collaborative filtering (Louppe, 2010).

Acknowledgments

GL and PG are respectively research fellow and research associate of the FNRS, Belgium. This paper presents research results of the Belgian Network BIOMAGNET (Bioinformatics and Modeling: from Genomes to Networks), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy.

References

Gimpel, K., Das, D., & Smith, N. (2010). Distributed asynchronous online learning for natural language processing. *Proceedings of the Conference on Computational Natural Language Learning.*

Louppe, G. (2010). Collaborative filtering: Scalable approaches using restricted Boltzmann machines. Master's thesis, University of Liège.

Nedic, A., Bertsekas, D., & Borkar, V. (2001). Distributed asynchronous incremental subgradient methods. *Studies in Computational Mathematics*, *8*, 381–407.

Salakhutdinov, R., Mnih, A., & Hinton, G. E. (2007). Restricted Boltzmann machines for collaborative filtering. *Proceedings of the 24th international conference on Machine learning* (p. 798).

Zinkevich, M., Smola, A., & Langford, J. (2009). Slow learners are fast. In *Advances in neural information processing systems 22*, 2331–2339.

Protein Subfamily Identification using Clustering Trees

Keywords: protein function prediction, phylogenomic analysis, decision trees, clustering trees

Eduardo P Costa[1], **Celine Vens**[1], **Hendrik Blockeel**[1,2]
{eduardo.costa, celine.vens, hendrik.blockeel}@cs.kuleuven.be
[1]Dept. of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, Leuven, Belgium
[2]Leiden Inst. of Advanced Computer Science, Universiteit Leiden, Niels Bohrweg 1, Leiden, The Netherlands

1. Introduction

One of the biggest challenges biologists are facing in the post-genomic era is the characterization of gene function. As this is a laborious and complex task, several computational methods have been designed to assist protein function prediction (Friedberg, 2006). The standard approach for protein function prediction is based on homology. To predict the function of a protein, homology-based methods look for a significantly similar sequence whose function is already characterized. If such a sequence can be found, its annotation is transferred to the new protein. Even though this procedure is based on the biological assumption that similar sequences are likely to have evolved from a common ancestor and have similar or identical functions, there are many exceptions (Whisstock & Lesk, 2003). In convergent evolution, for example, non-homologous protein sequences present similar functions. Other drawbacks of the homology-based approach are (1) the propagation of erroneous annotations throughout databases and (2) the lack of sensitivity of the current methods to deal with the growing in size and in diversity of the protein databases (Friedberg, 2006).

As an alternative to homology-based methods, phylogenomic analysis has been used for protein function prediction (Eisen, 1998; Brown et al., 2007). This analysis consists of using phylogenetic information to genomic studies. One of the tasks performed in this context is the protein subfamily identification. In this task one is interested in identifying subgroups of more closely related proteins within a family. Being able to identify these subfamilies is an important step in protein function prediction because such proteins usually share a specific function that is not common to the entire family. Current phylogenomic methods for protein subfamily identification first build a phylogenetic tree for proteins within a family, then extract clusters from it that hopefully correspond to subfamilies.

We propose a novel phylogenomic method that differs from the existing methods in two important ways: (1) it builds the phylogenetic tree top-down, rather than bottom-up, and stops when clusters are found; thus it avoids constructing the whole tree; (2) it associates particular mutations with each division into subclusters, allowing easy classification of new proteins.

2. Method

The proposed method is built on an existing decision tree learner, CLUS (Blockeel et al., 1998), which is based on divisive conceptual clustering that extends the well-known decision tree learning approach. Starting from a single cluster containing all sequences of a protein family, our method repeatedly divides it into subclusters until a set of clusters that potentially correspond to protein subfamilies is found.

We use a multiple sequence alignment as input to our method. This allows us to use polymorphic positions as tests to split sequences. As the number of splits based on these positions is linear in the length of the sequences, and constant in the size of the set, such a divisive method is not only computationally feasible, but also potentially faster than agglomerative methods. A test checks for the occurrence of an amino acid or a set of them at a particular location. The test "p5 $\in \{P, G\}$", for instance, creates two subsets, one containing all sequences with a P or G at position 5 and one containing all other sequences.

To choose the best split at each iteration of the divisive clustering procedure, we use a heuristic that can be seen as a top-down counterpart of the one used by the the well-known phylogenetic method Neighbor Joining - NJ (Saitou et al., 1987). The heuristic is based on the principle of minimum evolution, i.e. it aims at finding the tree with minimum total branch length (for more details about the heuristic, see (Vens et al., 2010)).

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the author(s)/owner(s).

At each step of the process, our method estimates the total branch length of the tree that will finally be constructed, for all tests being evaluated; then it chooses the best test according to these estimations.

Finally, there must be a stop criterion that defines when the final clusters, which will be predicted as protein subfamilies, have been found. For this, different heuristics can be used, such as when no more significant reduction of intracluster variance can be achieved. We are currently working on the stop criterion.

3. Experiments

We have tested our method on the eight EXPERT datasets proposed in (Brown et al., 2007). Three scenarios were designed for our experiments. In the first two scenarios we check individual aspects of our method: (1) the assumption that protein subfamilies can be discriminated by a decision tree that uses polymorphic positions as tests; (2) the heuristic to induce the decision tree. In the third scenario we test our method for the task of protein subfamily identification. We compare our results with SCI-PHY (Brown et al., 2007), which is the state of the art for protein subfamily identification using phylogenomic analysis.

Scenario 1. To check that trees that have splits based on polymorphic positions are useful for protein subfamily identification, we added the subfamily information to the data and induced a classification tree using CLUS (i.e. we performed a supervised classification task). The results showed that subfamilies can be perfectly separated from one another using compact trees containing slightly more leaves than the number of subfamilies in the datasets. For two of the datasets the classification tree has the same number of leaves as the number of subfamilies. This shows that the solution we are looking for is part of our search space.

Scenario 2. To evaluate the quality of the trees being produced, regardless of the stop criterion, we grew the tree completely until each node was a singleton, and then cut the tree in a way that all clusters were pure and that the tree was as compact as possible. We did the same with the trees produced by NJ and SCI-PHY. Our method produced more compact trees than NJ for five datasets, and more compact trees than SCI-PHY for seven datasets. This shows that our method can yield good results if an adequate stop criterion is used.

Scenario 3. In this scenario, which corresponds to the prediction problem we are interested in, we test the whole procedure, including a stop criterion and without knowledge of the protein subfamily classification of the sequences. We defined as stop criterion the point where the entropy reduction given by best test, according to the minimal total branch length heuristic, is less than five percent. The results showed that only for one dataset our method produced better results than SCI-PHY. For the other datasets the results were reasonably worse, showing that the chosen stop criterion was not adequate to define the right points to stop the growing of the tree. We observed that for most of the cases the tree stopped growing too soon. Given these results, we are now investigating new possibilities to define the stop-criterion.

4. Final Considerations

The proposed method has important advantages: (1) it is more efficient, since the clustering is performed top-down; (2) as each division into subclusters is defined by polymorphic locations, the resulting tree immediately gives an evolutionary trace, which can be useful for recognizing functional sites; (3) the induced decision tree can be used to classify new sequences. Results have shown that polymorphic positions can be used as tests to discriminate among protein subfamilies and that the phylogenetic construction module of our method produces trees of good quality. However, we still need to work further on the stop criterion for the final prediction of the protein subfamilies.

References

Blockeel, H., De Raedt, L., & Ramon, J. (1998). Top-down induction of clustering trees. *Proc. of the 15th International Conference on Machine Learning.*

Brown, D., Krishnamurthy, N., & Sjolander, K. (2007). Automated protein subfamily identification and classification. *PLoS Comput Biol*, *3*, 1526–1528.

Eisen, J. (1998). Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome research*, *8*, 163–167.

Friedberg, I. (2006). Automated protein function prediction: the genomic challenge. *Briefings in bioinformatics*, *7*, 225–242.

Saitou, N., Nei, M., et al. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, *4*, 406–425.

Vens, C., Costa, E. P., & Blockeel, H. (2010). Top-down induction of phylogenetic trees. *Lecture Notes in Computer Science,* (pp. 62–73). Springer.

Whisstock, J., & Lesk, A. (2003). Prediction of protein function from protein sequence and structure. *Quarterly reviews of biophysics*, *36*, 307–340.

Mining Closed Strict Episodes

Nikolaj Tatti NIKOLAJ.TATTI@UA.AC.BE
Boris Cule BORIS.CULE@UA.AC.BE
University of Antwerp, Middelheimlaan 1, 2020 Antwerpen, Belgium

1. Extended Abstract

Discovering frequent patterns in a sequence is an important field in data mining. Such a pattern is usually considered to be a set of events that reoccurs in the sequence within a window of a specified length. Gaps are allowed between the events and the order in which they occur is also considered important. Frequency, the number of windows in which the episode occurs, is monotonically decreasing so we can use the well-known WINEPI [1] method, a level-wise approach, to mine all frequent episodes. The order restrictions of an episode are described by a directed acyclic graph (DAG): the set of events in a sequence covers the episode if and only if each event occurs only after all its parent events (with respect to the DAG) have occurred.

Formally, a *sequence* $s = (s_1, \ldots, s_L)$ is a string of symbols coming from an *alphabet* Σ, so that for each i, $s_i \in \Sigma$. An episode G is represented by an acyclic directed graph with labelled nodes, that is, $G = (V, E, lab)$, where $V = (v_1, \ldots, v_K)$ is the set of nodes, E is the set of directed edges, and lab is the function $lab : V \to \Sigma$, mapping each node v_i to its label. We denote the set of nodes of an episode G with $V(G)$, and its set of edges with $E(G)$. Given a sequence s and an episode G we say that s *covers* G if there is an *injective* map f mapping each node v_i to a valid index such that the node v_i in G and the corresponding sequence element $s_{f(v_i)}$ have the same label, $s_{f(v_i)} = lab(v_i)$, and that if there is an edge (v_i, v_j) in G, then we must have $f(v_i) < f(v_j)$. In other words, the parents of v_j must occur in s before v_j.

Usually, only two extreme cases are considered. A parallel episode poses no restrictions on the order of events, and a window covers the episode if the events occur in the window, in any order. In such a case, the DAG associated with the episode contains no edges. The other extreme case is a serial episode. Such an episode requires that the events occur in one, and only one, specific order in the sequence. Clearly, serial episodes are more restrictive than parallel episodes. If a serial episode is frequent, then its parallel version is also frequent. General episodes have, in practice, been over-shadowed by parallel and serial episodes, despite being defined at the same time [1]. The main reason for this is the pattern explosion demonstrated in the following example. We used the inaugural speeches by presidents of the United States (taken from `http://www.bartleby.com/124/pres68`), and merged them to form a single long sequence of stemmed words. By setting the window size to 15 and the frequency threshold to 60 we discovered a serial episode with 6 symbols,

$$\text{preserv} \to \text{protect} \to \text{defend} \to \text{constitut} \to \text{unit} \to \text{state}.$$

In total, we found 4824 subepisodes of size 6 of this episode. All these episodes had only 3 distinct frequencies, so the frequencies of most of them could be derived from the frequencies of only 3 episodes, so 4821 episodes could safely be left out of the output.

Motivated by this example, we approach the problem of pattern explosion by using a popular technique of closed patterns. A pattern is closed if there exists no more specific pattern with the same frequency. Mining closed patterns has been shown to reduce the output. Moreover, if we can establish a specific property called the Galois connection, we can discover closed patterns efficiently. However, adopting the concept of closedness to episodes is not without problems.

Subset relationship Firstly, in order to define closed patterns we need a subset relationship between patterns to describe whether a pattern G is a subpattern of pattern H. Essentially the same episode can be described by multiple DAGs and if we would base our definition of closedness simply on a subset relationship of DAGs we will run into problems as demonstrated in the following example. Consider episodes G_1, G_2, and G_3 given in Figure 1. Episode G_1 states that for a pattern to occur a must precede b and c. G_2 and G_3, meanwhile, state that a must be followed by b and then

Appearing in Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands. Copyright 2011 by the author(s)/owner(s).

by c. Note that G_2 and G_3 represent essentially the same pattern that is more restricted than the pattern represented by G_1. However, G_1 is a subgraph of G_3 but not a subgraph of G_2. This reveals a problem if we base our definition of a subset relationship of episodes solely on the edge subset relationship. We solve this by generating only transitively closed graphs, thus ignoring graphs of form G_2. We will not lose any generality since we are still going to discover episodes of form G_3.

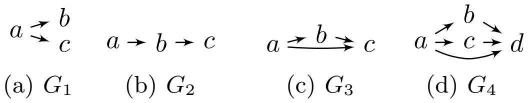


Figure 1. Examples of various episodes.

Frequency closure Secondly, frequency does not satisfy the Galois connection. In fact, given an episode G there can be *several* more specific closed episodes that have the same frequency. So the closure operator cannot be defined as a mapping from an episode to its frequency-closed version. Consider sequence $s = abcbdacbcd$ and episode G_4 given in Figure 1(d). Assume that we use a sliding window of size 5. There are two windows that cover episode G_4, namely $s_1 \cdots s_5$ and $s_6 \cdots s_{10}$. Hence, the frequency of G_4 is 2. There are *two* serial episodes that are more specific than G_4 and have the same frequency, namely, $H_1 = (a \rightarrow b \rightarrow c \rightarrow d)$ and $H_2 = (a \rightarrow c \rightarrow b \rightarrow d)$. Moreover, there is no superepisode of H_1 and H_2 that has frequency 2. Therefore, we cannot define a unique closure for G_4.

The contributions of our paper address these issues:

1. We introduce *strict* episodes, a new subclass of general episodes. Formally, an episode G is called *strict* if for any two nodes v and w in G sharing the same label, there exists a path either from v to w or from w to v. This class is large, contains all serial and parallel episodes, and most of the general episodes, yet using only strict episodes eases the computational burden.

2. We introduce a natural subset relationship between episodes based on the subset relationship of sequences covering the episode. We say a transitively closed episode G is called a *subset* of a transitively closed episode H, denoted $G \preceq H$ if the set of all sequences that cover H is a subset of the set of all sequences that cover G. If G is a proper subset of H, we denote $G \prec H$. If $|V(G)| < |V(H)|$, then G is a subset of H if there is a subgraph H' of H such that $G \preceq H'$. We prove that for strict episodes this relationship corresponds to the subset relationship between transitively closed graphs. For strict episodes such a graph uniquely defines the episode.

3. We introduce a milder version of the closure concept called the *instance-closure*. We begin by associating each sequence with a corresponding serial episode. Given a sequence $s = (s_1, \ldots, s_N)$, we define its *corresponding serial episode* G_s as the transitive closure of
$$\left(v_{m(1)} \rightarrow v_{m(2)} \rightarrow \cdots \rightarrow v_{m(N)}\right),$$
with $lab\left(v_{m(i)}\right) = s_i$. Mapping m makes sure that the nodes of G_s are ordered, that is, for $i < j$, $lab(v_i) \leq lab(v_j)$. We then build a maximal episode that is covered by a set of sequences. Given a set of nodes V, and a set S of sequences containing the events corresponding to labels of nodes in V, we define the *maximal episode* covered by S as the episode H, where $V(H) = V$ and $E(H) = \bigcap_{s \in S} E(G_s)$. We can make a Galois connection between the set of all episodes with a fixed set of nodes V and the power set $\mathcal{S}$ containing all sets of subsequences consisting only of labels of nodes V in all windows of length k in our sequence s, where k is the chosen window size. For all episodes G containing nodes V, we define
$$f(G) = \{\, w \mid w \in \mathcal{S},\, w \text{ covers } G \,\}.$$
For all sets of subsequences S in $\mathcal{S}$, we define a function g as $g(S) = G$, with G the maximal episode covered by S. This closure satisfies the Galois connection. Crucially, we note that a frequency-closed episode is always instance-closed.

4. Finally, we present an algorithm that generates strict instance-closed episodes with transitively closed graphs. Once these episodes are discovered we can further prune the output by removing the episodes that are not frequency-closed. The details of the algorithm and the complete experimental results are omitted here due to space constraints, but can be found in the full paper [2], along with a complete list of references.

References

[1] Mannila, H., Toivonen, H., & Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, *1*, 259–289.

[2] Tatti, N., & Cule, B. (2010). Mining closed strict episodes. *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM 2010).*

DESYDE: Decentralized (De)synchronization in Wireless Sensor Networks

Mihail Mihaylov[1] MMIHAYLO@VUB.AC.BE
Yann-Aël Le Borgne[1] YLEBORGN@VUB.AC.BE
Karl Tuyls[2] K.TUYLS@MAASTRICHTUNIVERSITY.NL
Ann Nowé[1] ANN.NOWE@VUB.AC.BE
[1]Vrije Universiteit Brussel, Pleinlaan 2, Brussels, Belgium
[2]Maastricht University, Sint Servaasklooster 39, Maastricht, The Netherlands

Keywords: multiagent learning, collective intelligence, teamwork, coalition formation, coordination, implicit cooperation, emergent behavior

Abstract

In the full version of this paper (Mihaylov et al., 2011) we propose DESYDE: a decentralized approach for coordinating the radio activity of wireless sensor nodes. Inspired by the *win-stay lose-shift* strategy from game theory, our approach allows individual nodes to schedule their radio transmission, reception and sleeping periods without any form of explicit coordination. We implement DESYDE in the OMNeT++ sensor network simulator and compare its performance to two state-of-the-art scheduling protocols, namely S-MAC and D-MAC. We show that our approach adapts the wake-up cycle of each node to its traffic load and significantly reduces end-to-end communication delays.

1. Introduction

Wireless Sensor Networks (WSNs) are a recent class of networks able to monitor our daily environment with a high spatiotemporal accuracy (Ilyas & Mahgoub, 2005). WSNs are composed of small sensing devices, also known as wireless sensor nodes, endowed with sensing, processing and wireless communication capabilities. The limited resources of the sensor nodes make the design of a WSN application challenging. Application requirements, in terms of latency, data throughput, or lifetime, often conflict with the network capacity and energy resources. The standard approach for addressing these tradeoffs is to rely on *wake-up scheduling* (Ilyas & Mahgoub, 2005), which consists in alternating the active and sleep states of sensor nodes. The fraction of time in which the node is in the active mode is referred to as *duty cycle*. The period of this cycle is called *frame*, which can be further divided into a number of *time slots*.

Wake-up scheduling offers an efficient way to significantly improve the lifetime of a WSN application, and is well illustrated by S-MAC, a standard synchronized medium access control (MAC) protocol for WSN (Ye et al., 2004). In S-MAC, the duty-cycle is fixed by the user, and all sensor nodes synchronize in such a way that their active periods take place at the same time. This synchronized active period enables neighboring nodes to communicate with one another.

In the full version of this paper (Mihaylov et al., 2011) we demonstrate how the performance (in terms of lifetime and data latency) of a WSN network can be further improved, if nodes not only synchronize, but also *desynchronize* with one another. More precisely, the duty cycles of nodes that need to communicate with one another are synchronized to improve message throughput. We say that those nodes belong to one *coalition*. At the same time, the schedules of groups of nodes which do not need to communicate are desynchronized in order to avoid radio interferences and packet losses. We refer to this type of coordination for short as *(de)synchronization*.

Our approach allows sensor nodes to coordinate their activities in a decentralized manner, by relying on the win-stay lose-shift (WSLS) strategy drawn from game

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the author(s)/owner(s).

theory (Posch, 1999). We call the approach DESYDE, which stands for DEcentralized SYnchronization and DEsynchronization.

2. DESYDE

In DESYDE, the coordination is achieved by rewarding successful interactions (e.g., acknowledged transmission) and penalizing the ones with a negative outcome (e.g., message loss or overhearing). This behavior drives the sensor nodes to repeat actions that result in positive feedback more often and to decrease the probability of unsuccessful interactions. Nodes that tend to select the same successful action naturally form a coalition. The main benefit of the proposed approach is that global (de)synchronization emerges from simple and local interactions without the need of central mediator or any form of explicit coordination.

We implement DESYDE in the OMNeT++ simulator (www.omnetpp.org), and study its performance in terms of lifetime and data latency for a data collection task, in which all nodes periodically report their data to a base station. We consider three different wireless sensor network topologies, namely line, grid, and random, and assume that the data are relayed from the nodes to the base station by means of a routing tree. We compare DESYDE to S-MAC (Ye et al., 2004) and D-MAC (Lu et al., 2004), two state-of-the-art coordination mechanisms for WSNs, and show that nodes form coalitions which improve data communication and reduce packet collisions. This enables a quicker delivery of the data packets to the base station, allowing shorter active periods. Comparing to the aforementioned protocols, we measured up to 50% lower energy consumption and a 10-fold reduction in latency on 50-node random topologies. On the grid topologies DESYDE achieves the low latency of S-MAC with 5 times lower duty cycle.

These improvements are obtained thanks to the (de)synchronization of the sensor nodes' schedules. We provide in Figure 1 an example of the schedules after applying DESYDE on a simple 2 by 2 grid topology. In this example, the frame contains 10 slots, and the four schedules reported are those of the four nodes in the grid, arranged in the same topological order. At slot 2, the upper left node transmits when the lower left node receives, while the right nodes are synchronized for communication at slot 5. The lower left node sends its data to the base station at slot 7 and forwards that of the upper left node at slot 9. The lower right node does the same at slots 4 and 6, respectively. Thus, we observe the same coalitions as in our schematic model in Figure 1 (left).

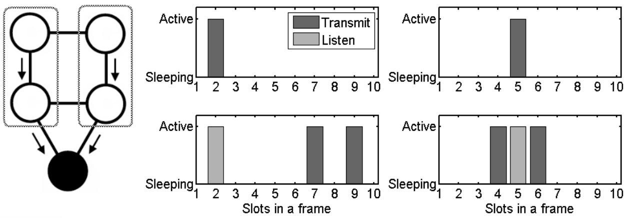


Figure 1. DESYDE

3. Conclusion

State-of-the-art synchronized protocols only perform well in simple networks, such as line topologies. As the network complexity grows, these protocols result in high latency and energy costs, due to the increased number of packet collisions and packet retransmissions. By mitigating these effects, DESYDE was able in all our experiments to compete with standard approaches, and exhibited significant gains in latency and energy especially for larger networks.

References

Ilyas, M., & Mahgoub, I. (2005). *Handbook of sensor networks: compact wireless and wired sensing systems.* CRC.

Lu, G., Krishnamachari, B., & Raghavendra, C. (2004). An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International* (p. 224).

Mihaylov, M., Le Borgne, Y.-A., Tuyls, K., & Nowé, A. (2011). Distributed cooperation in wireless sensor networks. *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)* (pp. 249–256). Taipei, Taiwan.

OMNeT++. http://www.omnetpp.org/ – a C++ simulation library and framework.

Posch, M. (1999). Win-Stay, Lose-Shift Strategies for Repeated Games–Memory Length, Aspiration Levels and Noise. *Journal of theoretical biology*, *198*, 183–195.

Ye, W., Heidemann, J., & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, *12*, 493–506.

Phenotype Classification of Zebrafish Embryos by Supervised Learning

Nathalie Jeanray[1], Raphaël Marée[2, 3], Benoist Pruvot[1], Olivier Stern[2], Pierre Geurts[2], Louis Wehenkel[2], Marc Muller[1]

Address: [1]GIGA-Development, Stem cells and regenerative medicine, Molecular Biology and Genetic Engineering, University of Liège, [2]GIGA-Systems Biology and Chemical Biology, Dept. EE & CS, University of Liège and [3]GIGA Bioinformatics Core Facility, University of Liège.

Keywords: Phenotype classification, supervised learning, zebrafish

Abstract

Zebrafish is increasingly used to assess biological properties of chemical substances and thus becomes a specific tool for toxicological and pharmacological studies. The effects of chemical substances on embryo development are generally evaluated manually through microscopic observation by an expert and documented by several typical photographs. Ideally, all the evaluated individuals should be photographed and classified according to the defects observed. Our project aims at reducing the workload and time required for the biological expert by automatic data acquisition through motorized microscopy, followed by classification of the obtained images. In order to increase the reproducibility and the objectivity of this classification, we present here a method to classify images of zebrafish embryos according to their defects automatically using a supervised learning approach. Automation of the analysis and classification of zebrafish pictures would become a real advantage for the biologists in terms of time and accuracy.

1. Introduction

Zebrafish, or "*Danio rerio*", is commonly used as a vertebrate model organism in the fields of develop[1]mental biology, but also increasingly in toxicology and pharmacology. Due to several advantages such as fast growth, *ex vivo* development, larvae transparency, low cost and permeability to small molecules, zebrafish appears as a powerful model to assess toxic activities on vertebrates. Typically, embryos are collected one by one and observed manually in order to detect potential developmental or morphological modifications. These modifications are listed, illustrated by a picture and statistically processed to infer the toxicological effects of the drug and the role it might play within the metabolic reactions. This method is tedious, time-consuming and prone to appreciation subjectivity. The large number of substances to be tested and the need for accuracy of the results call for methods allowing automation of both data acquisition and classification of the images.

2. Methods

2.1 Treatments

Zebrafish embryos were treated at 2 days post fertilization in batches of 25 individuals and analyzed after 24 hours of treatment. At this stage (3 days old), the embryos have hatched and are easily observable. Untreated control batches received only the solvent used for the drug stock solution.

2.2 Data Acquisition

In order to develop an accurate and non-biased approach to classify the pictures, a particular attention must be given to image acquisition. Photographs have been taken on an Olympus stereo dissecting microscope with the same parameters from one acquisition to the next (exposure time = 10ms, contrast = 1.05, maximum luminosity, white balance, magnification = 1.60x) to limit bias errors due to a non homogenous background. Embryos are placed in a melt of E3 and methylcellulose in a 12-well plate, one fish per well. The pictures are taken manually at this stage of the work.

2.3 Automated recognition pipeline

2.3.1 Image Pre-processing

Embryo images are first standardized based on three major steps. First, we apply a rotation in order to place all the embryos in the same position, which is, horizontally and head to the left. This step is performed automatically by an algorithm searching the position of the eyes and rotating the fish when needed. Secondly, the dorso-ventral orientation of the fish is determined

Appearing in Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands. Copyright 2011 by the author(s)/owner(s).

by a supervised classifier trained with dorsal and ventral examples, then we applied a flipping operation if needed. The third step consists of cropping the fish using connected component labeling.

Further pre-processing steps have been evaluated depending on the sought defect. In the case of pericardial edemas, we tested automatic cropping of the fish to keep only the anterior ventral part. For the curved tail, we applied an algorithm to cut off the head of the fish to concentrate on the trunk of the fish.

2.3.2 IMAGE LABELING

After standardization, images of embryos were labeled by three biologists working independently. The goal is to identify two types of malformations on embryos, and unaffected ones. The embryos observed have been treated with a medication at different concentrations. Most of these embryos developed some abnormal phenotypes as, for instance, pericardial edemas, curved tails and growth delay. We chose to focus our analyses on two deformations at first, pericardial edema and curved tail. Examples are shown in Figure 1.

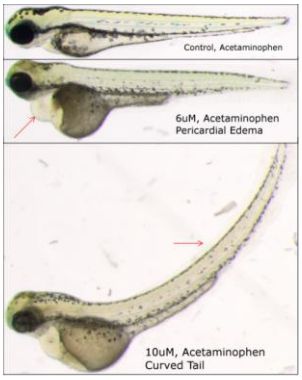


Figure 1. Illustration of the phenotypes analyzed in this study. Red arrows indicate the phenotypes studied. The top picture is a normal fish, the middle one presents a pericardial edema while the bottom image shows a curved tail.

2.3.3 SUPERVISED IMAGE CLASSIFICATION

Given the set of training images where each image was labeled into the majority class assigned by the three experts, the goal is to build a model by supervised learning that will be able to predict accurately the class of new, unseen images. We used the image classification algorithm of (Marée et al., 2007), a generic method which has been validated on many problems before envisaging the development of a more specific method. It is based on dense random subwindow extraction in images, their description by raw pixel values, and the use of ensembles of extremely randomized trees (Geurts et al., 2006) to classify these subwindows hence images.

3. Results

We evaluated our method on images acquired in three independent experiments with different numbers of images per class for each experiment (Figure 2).

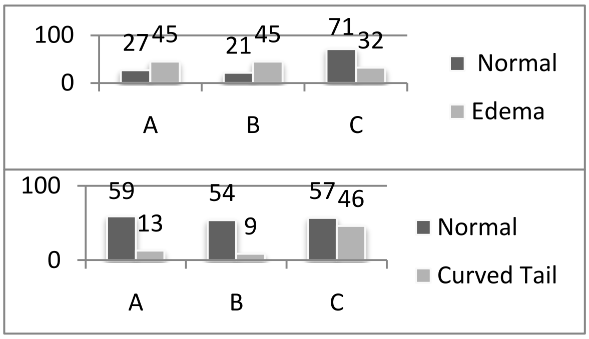


Figure 2. Number of images in each class (normal, edema or curved tail) in the three experiments A, B and C.

For each experiment and each deformation, we evaluate empirically the influence of the main parameters of the algorithm by cross-validation or leave-one-out protocols. Table 1 shows the best recognition rates we obtained. The parameters we modified in order to get these results were mainly the size ranges of the extracted subwindows, the classification scheme, the number of trees to build, the stop criterion based on the minimum node sample size, the number of random tests and the number of subwindows extracted within each image. Binary models are then built since we try to classify larvae presenting an edema vs. control larvae (edema vs. normal), and larvae with a curved tail vs. normal ones (curved tail vs. normal).

Table 1. Classification accuracies for “Edema” and “Curved Tail” classes.

DATA SET	A	B	C
EDEMA	96.0	59.9	98.4
CURVED TAIL	87.5	75.8	94.9

4. Conclusion and Perspectives

Our automatic classification method already gives promising results in the analysis of two different defects, edema and curved tails, allowing to anticipate that other morphological abnormalities could also be classified. In the future, we will focus on rendering the acquisition procedure fully automatic, while also extending our classification method to other defects.

References

Marée, R., Geurts, P., Wehenkel, L. (2007). *Random subwindows and extremely randomized trees for images classification in cell biology*. BMC Cell Biology, **8**:S2

Geurts P, Ernst D, Wehenkel L. (2006). *Extremely Randomized Trees*. Machine Learning, **36**:3-42

Zebrafish Skeleton Measurements using Image Analysis and Machine Learning Methods

Stern O.[1], Marée R.[1,2], Aceto J.[3], Jeanray N.[3], Muller M.[3], Wehenkel L.[1] and Geurts P.[1]
{OLIVIER.STERN, P.GEURTS}@ULG.AC.BE
[1]GIGA-Systems Biology and Chemical Biology, Dept. EE & CS, University of Liège; [2]GIGA Bioinformatics Core Facility, University of Liège; [3]GIGA-Development, Stem cells and regenerative medicine, Molecular Biology and Genetic Engineering, University of Liège

Keywords: machine learning, image segmentation, pixel annotation, zebrafish morphometry

Abstract

The zebrafish is a model organism for biological studies on development and gene function. Our work aims at automating the detection of the cartilage skeleton and measuring several distances and angles to quantify its development following different experimental conditions.

1. Introduction

In this work, we address two subproblems: 1) quantifying the surface of the cartilage skeleton and 2) detecting several points of interest in zebrafish images. These two problems are not trivial from a machine learning point of view. These are essentially structured output problems for which there exist no straightforward solution. In addition, labels have to be indicated manually and thus training sets are rather small. Finally, these annotation problems are not easy even for human.

2. Image Generation

As we focus on the evolution of the cartilage skeleton, zebrafish larvae were grown fom 4 to 10 days post-fertilization, sacrificed by tricaine treatment and fixed in 4% para-formaldehyde. Alcian blue staining of the cartilage extracellular matrix was performed according to (Kimmel et al., 1998) and photographs were obtained using a Olympus dissection microscope by observing single larvae under identical conditions. Ventral and lateral views were recorded, but in this ongoing work, we will only use 15 ventral views.

3. Segmentation of Cartilage Skeleton

Our goal is to isolate the cartilage skeleton to quantify its surface. This approach uses supervised learning methods to partition the elements of the embryo within a new, unseen image by predicting the class of every pixel. Our starting point is a technique developed in (Dumont et al., 2009) which extracts a random sample of subwindows in a set of images with pixel-wise labelling (i.e. every pixel is labelled with one class among a finite set of predefined classes) and exploits extremely randomized trees to build a classifier.

Training Stage. We assume a partial annotation by an expert of the image pixels into three classes: Eye, Skeleton and Others. For each image of the learning sample, we construct subwindows of size w x h centered on the labelled pixels. Subwindows are then tagged by the class of the central pixel (output) and described by the color (i.e. the values in HSV and RGB color spaces), the edges (i.e. the gradient of the Sobel operator) and the texture (i.e. the histogram of local binary pattern (Ojala et al., 1996)) of the pixels in the subwindows (inputs). From this learning set, we construct a classification model using extremely randomized trees.

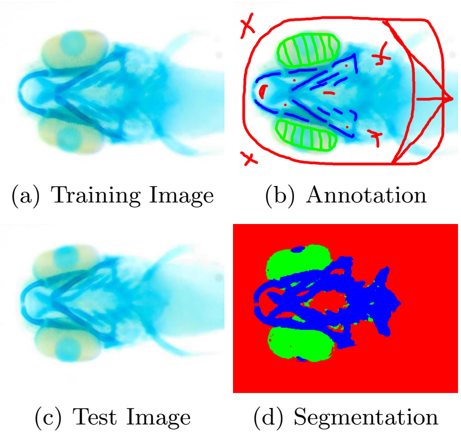
Prediction Stage. A subwindow centered on each pixel of every test image is extracted and its class is predicted using tree ensembles.

Parameters. The approach depends on several parameters that are related to the machine learning method (number of trees, number of random test splits) or to the subwindow descriptors (type of attributes, size of the subwindow). We only report our best results below.

Results. We manually and partially labelled 3 images out of the 15 (see Figures 1(a) and 1(b) for one ex-

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).

ample), from which the classification model was learnt. The 12 remaining images were then automatically annotated. Figures 1(c) and 1(d) show a test image and its annotation. The assessment of our model is difficult in the absence of a complete ground truth annotation but results seem visually very satisfying.


(a) Training Image (b) Annotation

(c) Test Image (d) Segmentation

Figure 1. Segmentation of Cartilage Skeleton

4. Detection of Points of Interest

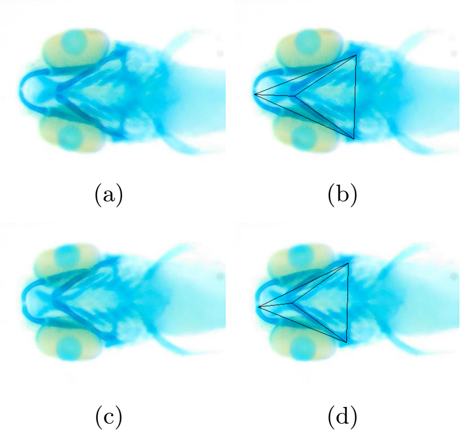
The objective of our second contribution is to automatically detect particular points of interest in the image, corresponding to particular position in the cartilage. By using several of such detectors, we can then automatically measure various lengths and angles defined by the corresponding points. As for pixel classification, we describe our pixel with a subwindow centered on the points of interest and we exploit extremely randomized trees to build a classifier.

Learning Stage. For a given point of interest, we indicate its coordinates in all training images. The learning set is then composed, on one hand, of subwindows centered on pixels in a radius r around the pixel of interest (positive class) and, on the other hand, of subwindows taken randomly in the rest of the image (negative class). Subwindows features are the same as in Section 3. Each particular point of interest thus requires the construction of a separate binary classification model using extremely randomized trees.

Prediction Stage. We predict the class of every subwindow centered on the pixels of the test images using tree ensembles. To determine the final coordinates of the points, we rank the prediction on their predicted probability and take either the mean, the median or a weighted mean of the coordinates of the best predicted pixels (i.e. above a given theshold).

Parameters. In addition to the parameters of the learning algorithm and subwindow descriptors, this method introduces several new parameters such as the radius around the points and the threshold on predicted probabilities. We tested several values of these parameters and only report the best results below.

Results. We apply this idea to detect four different points of interest in zebrafish images. These four points were manually identified in all images and then a leave-one-out was performed to assess the four classifiers. Figure 2 shows the prediction of these four points in two test images. Once these points have been detected several measures of lengths and angles can be done (represented by straight lines in Figures 2(b) and 2(d)).


(a) (b)

(c) (d)

Figure 2. Detection of points of interest

5. Future Work

We will further investigate ways to improve point detection, using multi-class classifiers and/or regression trees, and evaluate the approach on larger image sets of larvae presenting various defects in cartilage formation to obtain automatic determination of the different morphometric measures.

References

Dumont, M., Marée, R., Wehenkel, L., & Geurts, P. (2009). Fast multi-class image annotation with random subwindows and multiple output randomized trees. *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP).*

Kimmel, C. B., Miller, C. T., Kruze, G., Ullmann, B., BreMiller, R. A., Larison, K. D., & Snyder, H. C. (1998). The shaping of pharyngeal cartilages during early development of the zebrafish. *Dev Biol*, *203*, 245–263.

Ojala, T., Pietikainen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, *29*, 51–59.

Policy gradient methods for controlling systems with discrete sensor information

Matteo Gagliolo MGAGLIOL@VUB.AC.BE
Kevin Van Vaerenbergh
Abdel Rodríguez
Ann Nowé
CoMo, VUB (Vrije Universiteit Brussel), Pleinlaan 2, 1050 Brussels, Belgium

Stijn Goossens STIJN.GOOSSENS@FMTC.BE
Gregory Pinte
Wim Symens
FMTC (Flanders' Mechatronics Technology Centre), Celestijnenlaan 300D, 3001 Heverlee, Belgium

Abstract

In most existing motion control algorithms, a reference trajectory is tracked, based on a continuous measurement of the system's response. In many industrial applications, however, it is either not possible or too expensive to install sensors which measure the system's output over the complete stroke: instead, the motion can only be detected at certain discrete positions. The control objective in these systems is often not to track a complete trajectory accurately, but rather to achieve a given state at the sensor locations (e.g. to pass by the sensor at a given time, or with a given speed). Model-based control strategies are not suited for the control of these systems, due to the lack of sensor data. We are currently investigating the potential of a non-model-based learning strategy, Reinforcement Learning (RL), in dealing with this kind of discrete sensor information. Here, we describe experiments with a simple yet challenging system, where a single sensor detects the passage of a mass being pushed by a linear motor.

RL problems (Sutton & Barto, 1998) are a class of machine learning problems, where an agent must learn to interact with an unknown environment, using a "trial and error" approach. At a given timestep t, the agent may execute one of a set of *actions* $a \in \mathcal{A}$, possibly causing the environment to change its *state* $s \in \mathcal{S}$, and generate a (scalar) *reward* $r \in \mathbb{R}$. An agent is represented by a *policy*, mapping states to actions. The aim of a RL algorithm is to optimize the policy, maximizing the reward accumulated by the agent. Learning is organized in a sequence of *epochs*, each consisting of a sequence of interactions with the environment. Simply stated, RL consists in learning from a teacher (the environment) who cannot tell us *what* to do next (the optimal policy), but only *how good* we are doing so far (the reward signal). It therefore offers a suitable framework for the control of systems with discrete sensor information. The target state at the discrete sensors location can be incorporated in the reward signal, in order to favor the desired behavior.

The potential of different RL techniques is validated on a set-up consisting of a linear motor and a moving mass mounted on a straight horizontal guide (Figure 1). The position of the moving mass is monitored via a single discrete sensor, set along the guide, which fires at the passage of a small (1 cm) element attached to the mass. When the motor is activated, the mass is "punched" forward, and slides up to a certain position, depending on the duration and speed of the motor's stroke, and on the unknown friction among the mass and its guide.

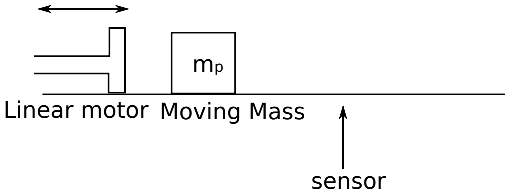


Figure 1: Sketch set-up, position measurement at one location

Two tasks are defined on this setup, with different objectives: a) let the mass pass the sensor at a predefined time

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the author(s)/owner(s).

(time task); b) let the mass stop exactly in front of the sensor (position task). As the motor movement precedes the passing of the sensor, conventional feedback control methods cannot obviously be applied to solve these tasks. For simplicity, we only consider constant speed signals, with duration varying on a closed interval, such that an action consists of a single scalar, which we normalize in $[0,1]$, and an epoch consists of a single action, followed by a scalar reward. For each task, a corresponding reward function is implemented, favoring the desired behavior[1]. Fig. 2 reports samples of the two reward functions: note that the system is highly stochastic, and repeating the same action twice can lead to different rewards.

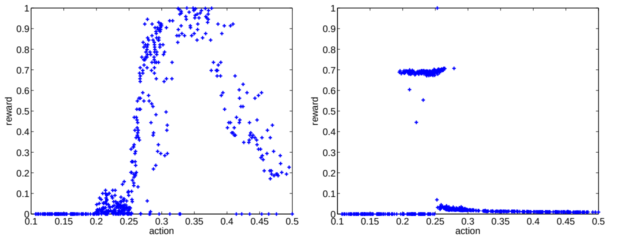


Figure 2: Reward functions for the two tasks, sampled for 500 randomly chosen actions on a subset of the unit interval.

Three *direct policy search* RL methods are evaluated on this setup, all representing the policy as a probability density function (pdf) over actions, which is updated offline, after each epoch: the policy gradient (PG) method (Peters & Schaal, 2006), where the actions are drawn from a Gaussian distribution; PG with parameter exploration (PGPE) (Sehnke et al., 2010), where the pdf is a mixture of Gaussians[2]; and a continuous learning automaton (CARLA) (Rodríguez et al., 2011), where the pdf over the actions is non-parametric.

While all three methods can successfully solve both tasks, CARLA displays faster convergence in both cases. Fig. 3, 4 report example runs on the two tasks. The position task turns out to be more difficult: this can easily be explained comparing the reward samples (Fig. 2). The best action for the position task, around 0.25, is a "needle in a haystack" compared to the time task, where the reward function changes more gradually around the optimal action.

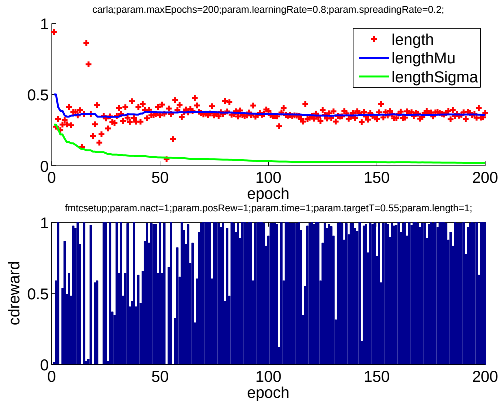


Figure 3: Time task, CARLA

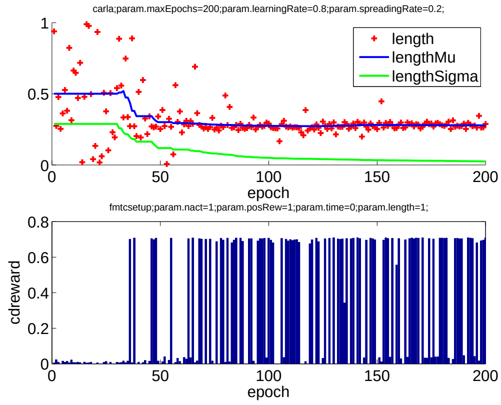


Figure 4: Position task, CARLA

Acknowledgment. This work has been carried out within the framework of the LeCoPro project (grant nr. 80032) of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

Peters, J., & Schaal, S. (2006). Policy gradient methods for robotics. *Proceedings of the IEEE Intl. Conf. on Intelligent Robotics Systems (IROS)*.

Rodríguez, A., Grau, R., & Nowé, A. (2011). Continuous actions reinforcement learning automata. performance and convergence. *Intl. Conf. on Agents and Artificial Intelligence (ICAART)*.

Sehnke, F., Osendorfer, C., Rückstieß, T., Graves, A., Peters, J., & Schmidhuber, J. (2010). Parameter-exploring policy gradients. *Neural Networks*, *23*, 551–559.

Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA, MIT Press.

[1] For the time task, given a target time t_0, reward is given as $r = \exp\{-c(t-t_0)^2\}$, where c is a constant, and t is the time at which the sensor starts firing, which is ∞ if the mass does not reach it. For the position task, the reward is given as the portion of time during which the sensor fires, over a constant time interval measured from the beginning of the motor's movement.

[2] This method is conceptually different from PG in that the pdf is not over actions, but over parameters of the policy, which are drawn at the beginning of an epoch. In this case, however, the action is a single parameter, and the epoch a single time step, so the two methods differ only for the pdf used (single Gaussian vs. mixture)

Flexible Enrichment with Cortana – Software Demo

Keywords: subgroup discovery, enrichment, software

Marvin Meeng MEENG@LIACS.NL
LIACS, Leiden University, Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands

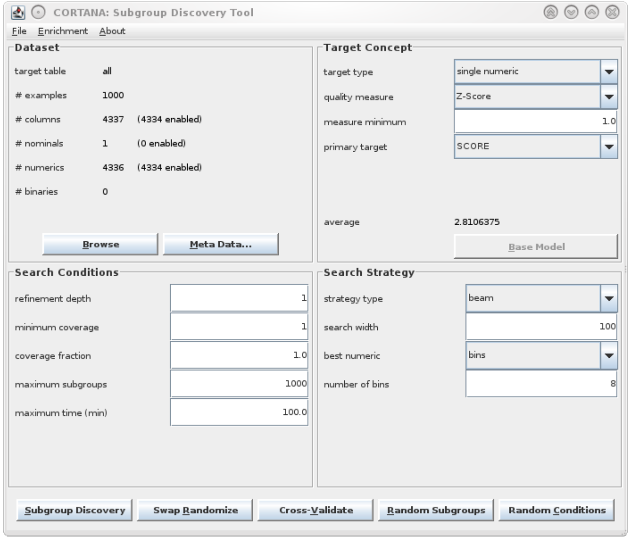
Arno Knobbe KNOBBE@LIACS.NL
LIACS, Leiden University, Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands

Abstract

The software demonstration will introduce an open-source package, called *Cortana*, that simplifies and unifies the procedure of enriching a list, or ranking, of biological entities with background knowledge. A host of software applications already exists that can do this enrichment [2, 5, 9]. However, most are focused on enriching only a single kind of biological entity, eg. genes in the case of gene set enrichment [6], or they are concerned with only a single source of background knowledge, eg. biological processes from GO. As a result, these tools are by design limited in their ability to integrate background knowledge from a variety of domain-crossing sources. The software tool introduced here, *Cortana*, will allow integration of knowledge extracted from both existing sources, like the online knowledge bases [3, 4, 8] that are used by other enrichment tools, as well as custom made ones, created by, or available to, the end user.

1. Cortana - Main

The rationale behind using *Cortana* as the basis for the proposed enrichment procedure is that it is a generic data mining tool. As such, it benefits from recent developments in the data mining field. These include, for example, statistically sound validation methods and a range of well-understood, and well-tested quality measures. Finally, it can deal with a variety of data types, including nominal, numeric, ordinal and binary. Furthermore, as it is not purpose-build, it can be used for a wide variety of data mining tasks, biological enrichment being among them, and this allows for many different primary and background sources to be used.



Dataset
target table all
# examples 1000
# columns 4337 (4334 enabled)
# nominals 1 (0 enabled)
# numerics 4336 (4334 enabled)
# binaries 0

Target Concept
target type single numeric
quality measure Z-Score
measure minimum 1.0
primary target SCORE
average 2.8106375

Search Conditions
refinement depth 1
minimum coverage 1
coverage fraction 1.0
maximum subgroups 1000
maximum time (min) 100.0

Search Strategy
strategy type beam
search width 100
best numeric bins
number of bins 8

Figure 1. Main window with parameters set for mining run.

2. Cortana - Bioinformatics facilities

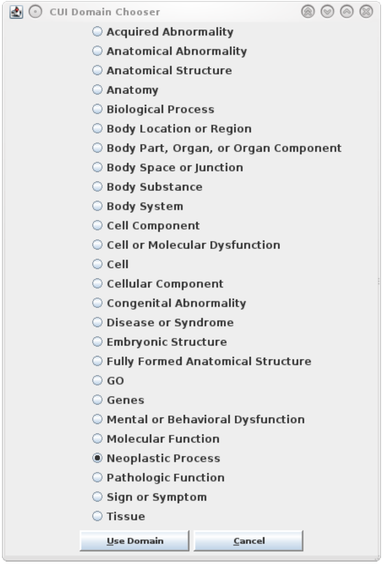
To smoothly integrate the enrichment functionality into *Cortana* a 'bioinformatics module' was added, allowing easy deployment of the tool for enrichment tasks. The typical workflow is described in the next section. The end user can use both existing and self defined, or custom, background sources. The tool is made available with domain files from two knowledge bases. The first consist of the three *GO*-domains: *biological process*, *cellular component* and *molecular function*, the well known binary gene/GO-term associations provided by `http://www.geneontology.org` [3]. The other is a set of so-called *association-matrices* derived from the literature by means of text mining. To

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.*

create these, a set of common concepts is determined through text mining of a large corpus of PubMed articles [1, 7], and then association scores between these concepts are calculated and stored in a matrix, or actually multiple matrices, each one specific to a certain domain. Alternatively, a user can use custom background sources. However, in this case, the end user will also need to supply a mapping file, allowing the entities in the input ranking to be matched to the entities in a background source.

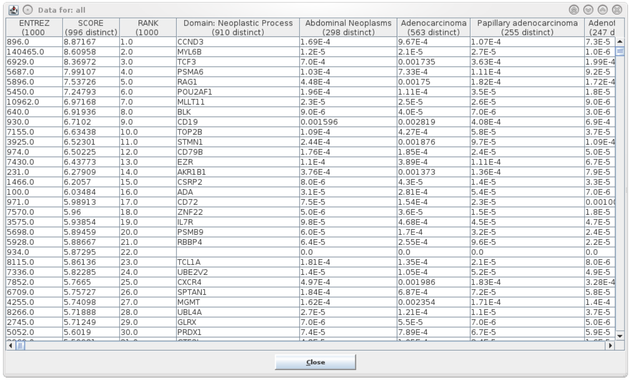
3. Cortana - Workflow

This section describes the typical workflow when enriching a (gene-)ranking with background knowledge. One starts by opening a file containing a ranking of biological entities. Typically this will be the result of an analysis of data obtained in a microarray experiment, yielding a list of *differentially expressed genes*, although other types of biological entities (eg. proteins) can be dealt with analogously. Figure 3 shows part of the ranking used in this example, with details such as the ENTREZ-identifier of the gene in question, its score resulting from the primary data analysis, its resulting rank, and the gene-symbol. Figure 1 shows *Cortana*'s main window, in which some information about the loaded data is displayed, such as the number of genes in the ranking, and the total number of descriptive values per gene that is available for enrichment. After loading the ranking, one or more files containing background knowledge from various domains can be included in the analysis. This is achieved by pressing 'Add CUI Domain', which will show a list of available CUI domains, as seen in Figure 2. Note that adding multiple background sources is possible. If one uses a custom background source, one will be asked to also indicate which file to use, to map the entities in the ranking to the corresponding information in the background source.



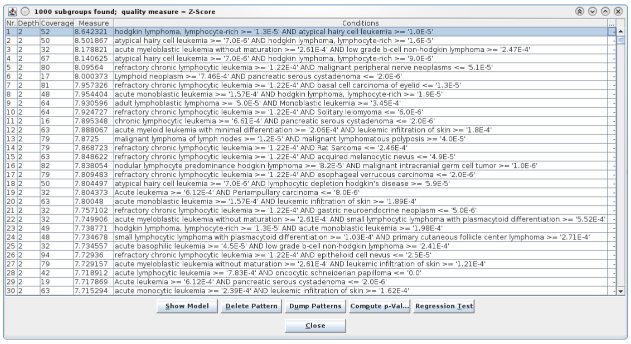
○ Acquired Abnormality
○ Anatomical Abnormality
○ Anatomical Structure
○ Anatomy
○ Biological Process
○ Body Location or Region
○ Body Part, Organ, or Organ Component
○ Body Space or Junction
○ Body Substance
○ Body System
○ Cell Component
○ Cell or Molecular Dysfunction
○ Cell
○ Cellular Component
○ Congenital Abnormality
○ Disease or Syndrome
○ Embryonic Structure
○ Fully Formed Anatomical Structure
○ GO
○ Genes
○ Mental or Behavioral Dysfunction
○ Molecular Function
● Neoplastic Process
○ Pathologic Function
○ Sign or Symptom
○ Tissue

Figure 2. Selecting a domain to add to ALL gene ranking.



ENTREZ (1000	SCORE (996 distinct)	RANK (1000	Domain: Neoplastic Process (910 distinct)	Abdominal Neoplasms (298 distinct)	Adenocarcinoma (563 distinct)	Papillary adenocarcinoma (255 distinct)	Adenol (247 d
896.0	8.87167	1.0	CCND3	1.69E-4	9.67E-4	1.07E-4	7.3E-5
140465.0	8.60958	2.0	MYL6B	1.2E-5	2.1E-5	2.7E-5	1.0E-6
6929.0	8.36972	3.0	TCF3	7.0E-4	0.001735	3.63E-4	1.99E-4
5687.0	7.99107	4.0	PSMA6	1.03E-4	7.33E-4	1.11E-4	9.2E-5
5896.0	7.53726	5.0	RAG1	4.48E-4	0.00175	1.82E-4	1.72E-4
5450.0	7.24793	6.0	POU2AF1	1.96E-4	1.11E-4	3.5E-5	1.8E-5
10962.0	6.97168	7.0	MLLT11	2.3E-5	2.5E-5	2.6E-5	9.0E-6
640.0	6.91936	8.0	BLK	9.0E-6	4.0E-5	7.0E-6	3.0E-6
930.0	6.7102	9.0	CD19	0.001596	0.002819	4.08E-4	6.9E-4
7155.0	6.63438	10.0	TOP2B	1.09E-4	4.27E-4	5.8E-5	3.7E-5
3925.0	6.52301	11.0	STMN1	2.44E-4	0.001876	9.7E-5	1.09E-4
974.0	6.50225	12.0	CD79B	1.76E-4	1.85E-4	2.4E-5	5.0E-5
7430.0	6.43773	13.0	EZR	1.1E-4	3.89E-4	1.11E-4	6.7E-5
231.0	6.27909	14.0	AKR1B1	3.76E-4	0.001373	1.36E-4	7.9E-5
1466.0	6.2057	15.0	CSRP2	8.0E-6	4.3E-5	1.4E-5	3.3E-5
100.0	6.03484	16.0	ADA	3.1E-5	2.81E-4	5.4E-5	7.0E-6
971.0	5.98913	17.0	CD72	7.5E-5	1.54E-4	2.3E-5	0.00100
7570.0	5.96	18.0	ZNF22	5.0E-6	3.6E-5	1.5E-5	1.8E-5
3575.0	5.93854	19.0	IL7R	9.8E-5	4.68E-4	4.5E-5	4.7E-5
5698.0	5.89459	20.0	PSMB9	6.0E-5	1.7E-4	3.2E-5	2.4E-5
5928.0	5.88667	21.0	RBBP4	6.4E-5	2.55E-4	9.6E-5	2.2E-5
934.0	5.87295	22.0		0.0	0.0	0.0	0.0
8115.0	5.86136	23.0	TCL1A	1.81E-4	1.35E-4	2.1E-5	8.0E-6
7336.0	5.82285	24.0	UBE2V2	1.4E-5	1.05E-4	5.2E-5	4.9E-5
7852.0	5.7665	25.0	CXCR4	4.97E-4	0.001986	1.83E-4	3.28E-4
6709.0	5.75727	26.0	SPTAN1	1.84E-4	6.87E-4	7.2E-5	5.8E-5
4255.0	5.74098	27.0	MGMT	1.62E-4	0.002354	1.71E-4	1.4E-4
8266.0	5.71888	28.0	UBL4A	2.7E-5	1.21E-4	1.1E-5	3.7E-5
2745.0	5.71249	29.0	GLRX	7.0E-6	5.5E-5	7.0E-6	5.0E-6
5052.0	5.6019	30.0	PRDX1	7.4E-5	7.89E-4	6.7E-5	5.9E-5

Figure 3. Browse combined table of ALL gene ranking and domain (*neoplastic process* in this case).


1000 subgroups found; quality measure = Z-Score

Nr.	Depth	Coverage	Measure	Conditions	...
1	2	52	8.642321	hodgkin lymphoma, lymphocyte-rich >= '1.3E-5' AND atypical hairy cell leukemia >= '1.0E-5'	-
2	2	50	8.501867	atypical hairy cell leukemia >= '7.0E-6' AND hodgkin lymphoma, lymphocyte-rich >= '1.6E-5'	-
3	2	32	8.178821	acute myeloblastic leukemia without maturation >= '2.61E-4' AND low grade b-cell non-hodgkin lymphoma >= '2.47E-4'	-
4	2	67	8.140625	atypical hairy cell leukemia >= '7.0E-6' AND hodgkin lymphoma, lymphocyte-rich >= '9.0E-6'	-
5	2	80	8.09564	refractory chronic lymphocytic leukemia >= '1.22E-4' AND malignant peripheral nerve neoplasms <= '5.1E-5'	-
6	2	17	8.000373	Lymphoid neoplasm >= '7.46E-4' AND pancreatic serous cystadenoma <= '2.0E-6'	-
7	2	81	7.957326	refractory chronic lymphocytic leukemia >= '1.22E-4' AND basal cell carcinoma of eyelid <= '1.3E-5'	-
8	2	48	7.954404	acute monoblastic leukemia >= '1.57E-4' AND hodgkin lymphoma, lymphocyte-rich >= '1.9E-5'	-
9	2	64	7.930596	adult lymphoblastic lymphoma >= '5.0E-5' AND Monoblastic leukemia >= '3.45E-4'	-
10	2	64	7.924727	refractory chronic lymphocytic leukemia >= '1.22E-4' AND Solitary leiomyoma <= '6.0E-6'	-
11	2	16	7.895348	chronic lymphocytic leukemia >= '6.61E-4' AND pancreatic serous cystadenoma <= '2.0E-6'	-
12	2	63	7.888067	acute myeloid leukemia with minimal differentiation >= '2.06E-4' AND leukemic infiltration of skin >= '1.8E-4'	-
13	2	79	7.8725	malignant lymphoma of lymph nodes >= '1.2E-5' AND malignant lymphomatous polyposis >= '4.0E-5'	-
14	2	79	7.868723	refractory chronic lymphocytic leukemia >= '1.22E-4' AND Rat Sarcoma <= '2.46E-4'	-
15	2	63	7.848622	refractory chronic lymphocytic leukemia >= '1.22E-4' AND acquired melanocytic nevus <= '4.9E-5'	-
16	2	82	7.838054	nodular lymphocyte predominance hodgkin lymphoma >= '8.2E-5' AND malignant intracranial germ cell tumor >= '1.0E-6'	-
17	2	79	7.809483	refractory chronic lymphocytic leukemia >= '1.22E-4' AND esophageal verrucous carcinoma <= '2.0E-6'	-
18	2	50	7.804497	atypical hairy cell leukemia >= '7.0E-6' AND lymphocytic depletion hodgkin's disease >= '5.9E-5'	-
19	2	32	7.804373	Acute leukemia >= '6.12E-4' AND Periampullary carcinoma <= '8.0E-6'	-
20	2	63	7.80048	acute monoblastic leukemia >= '1.57E-4' AND leukemic infiltration of skin >= '1.89E-4'	-
21	2	32	7.757102	refractory chronic lymphocytic leukemia >= '1.22E-4' AND gastric neuroendocrine neoplasm <= '5.0E-6'	-
22	2	32	7.749906	acute myeloblastic leukemia without maturation >= '2.61E-4' AND small lymphocytic lymphoma with plasmacytoid differentiation >= '5.52E-4'	-
23	2	49	7.738771	hodgkin lymphoma, lymphocyte-rich >= '1.3E-5' AND acute monoblastic leukemia >= '1.98E-4'	-
24	2	48	7.734678	small lymphocytic lymphoma with plasmacytoid differentiation >= '1.03E-4' AND primary cutaneous follicle center lymphoma >= '2.71E-4'	-
25	2	32	7.734557	acute basophilic leukemia >= '4.5E-5' AND low grade b-cell non-hodgkin lymphoma >= '2.41E-4'	-
26	2	94	7.72936	refractory chronic lymphocytic leukemia >= '1.22E-4' AND epithelioid cell nevus <= '2.5E-5'	-
27	2	79	7.729157	acute myeloblastic leukemia without maturation >= '2.61E-4' AND leukemic infiltration of skin >= '1.21E-4'	-
28	2	42	7.718912	acute lymphocytic leukemia >= '7.83E-4' AND oncocytic schneiderian papilloma <= '0.0'	-
29	2	19	7.717869	Acute leukemia >= '6.12E-4' AND pancreatic serous cystadenoma <= '2.0E-6'	-
30	2	63	7.715294	acute monocytic leukemia >= '2.39E-4' AND leukemic infiltration of skin >= '1.62E-4'	-

Figure 4. Result of enriching ALL gene ranking with Neoplastic Process.

Cortana will now combine the original data with the selected domain(s), the result of which can be checked by selecting 'Browse' (see Figure 3) or 'MetaData'. Then, one sets the parameters to be used for the mining run, and starts the actual enrichment by pressing 'Subgroup Discovery', again see Figure 1. After the mining finishes, a result window like Figure 4 will be shown, listing those subgroups, or concepts, that were found to be interesting, according to the selected parameters. In this case, setting the search parameter *refinement depth* to *2*, see Figure 1, allowed two conditions to be combined to form subgroups.

4. Benefits of the presented tool

Cortana is a flexible data mining tool for exploratory data analysis. With the addition of a bioinformatics module, its range of generic data mining facilities can now be employed for enrichment in the biological and medical domain. The tool comes with a large collection of background knowledge that can be involved in the enrichment process. This collection includes the customary functional annotations from GO (binary concepts), as well as the association matrices describing the level of association between each entity in the ranking and concepts from a number of domains (numeric concepts). In total, the 28 background files comprise 26.8 GB of background information available for enrichment. The bioinformatics module of *Cortana* has been employed in a number of medical and biological applications, including enrichment of neuroblastoma gene rankings, leukaemia (ALL and AML) gene rankings, and metabolic syndrome gene and metabolites rankings.

References

[1] Jelier, Schuemie, Veldhoven, et al., *Anni 2.0: a multipurpose text-mining tool for the life sciences*, Genome Biology 2008, 9(6):R96.

[2] Glynn Dennis Jr. et al., *DAVID: Database for Annotation, Visualization, and Integrated Discovery*, Genome Biology, 2003 4(9):R60, `http://david.abcc.ncifcrf.gov/home.jsp`.

[3] The Gene Ontology, 2011, `http://www.geneontology.org`.

[4] Ensembl, 2011, `http://www.ensembl.org`.

[5] Zeeberg B.R. et al., *GoMiner: A Resource for Biological Interpretation of Genomic and Proteomic Data*, Genome Biology, 2003 4(4):R28, doi:10.1186/gb-2003-4-4-r28, `http://discover.nci.nih.gov/gominer/index.jsp`.

[6] Subramanian, et al., *Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles*, PNAS October 25, 102:43, 2005.

[7] Jelier, Goeman, Hettne, Schuemie, den Dunnen & 't Hoen, *Literature-aided interpretation of gene expression data with the weighted global test*, Briefings in Bioinformatics, doi:10.1093/bib/bbq082, 2010.

[8] KEGG: Kyoto Encyclopedia of Genes and Genomes, 2011, `http://www.genome.jp/kegg/`.

[9] Search for Enriched Gene Sets, 2011, `http://kt.ijs.si/software/SEGS`.

[10] Trajkovski, Lavrač & Tolar, *SEGS: Search for enriched gene sets in microarray data*, Journal of Biomedical Informatics, 41(4), 588–601, 2008, `http://dx.doi.org/10.1016/j.jbi.2007.12.001`.

Cortana

Open Source Subgroup Discovery

Towards automatic monitoring of activities using contactless sensors

Marc Mertens MARC.MERTENS@KHK.BE
Glen Debard GLEN.DEBARD@KHK.BE
K.H. Kempen University College, MOBILAB, Kleinhoefstraat 4, 2440 Geel, Belgium
Jonas Van Den Bergh JONAS.VANDENBERGH@LESSIUS.EU
Toon Goedemé TGOEDEME@ESAT.KULEUVEN.BE
Lessius University College, Campus DE NAYER, Jan de Nayerlaan 5, 2860 Sint-Katelijne-Waver,Belgium
Koen Milisen KOEN.MILISEN@MED.KULEUVEN.BE
K.U.Leuven, Center for Health Services and Nursing Research, Kapucijnenvoer 35/4, 3000 Leuven, Belgium
K.U.Leuven, Division of Geriatric Medicine, Herestraat 49, 3000 Leuven, Belgium
Jos Tournoy JOS.TOURNOY@MED.KULEUVEN.BE
K.U.Leuven, Gerontology and Geriatrics section, Herestraat 49, 3000 Leuven, Belgium
Jesse Davis JESSE.DAVIS@CS.KULEUVEN.BE
K.U.Leuven, Department of Computer Science, Celestijnenlaan 200A ,3001 Heverlee, Belgium
Tom Croonenborghs TOM.CROONENBORGHS@KHK.BE
Bart Vanrumste BART.VANRUMSTE@KHK.BE
K.H. Kempen University College, MOBILAB, Kleinhoefstraat 4, 2440 Geel, Belgium

Keywords: ADL, contactless monitoring, sensor fusion, assisted living

Abstract

Due to the aging of the population, there will be more and more elderly who depend on the support of others. In order to increase the quality of care and life and to keep the cost of healthcare sustainable, we need to find ways to support these elderly to live independently in their own environment as long as possible in a comfortable way. This research aims at the non-intrusive monitoring of activities of daily living (ADL) of elderly people living alone at home. Identifying and labeling lifestyle patterns are carried out automatically and non-intrusively by using contactless sensoring and machine learning techniques. Sensor fusion is used to make the detection algorithm more robust. Sudden (e.g., accidents) or slowly changing (e.g., cognitive and behavioral disturbances in dementia) deviations from these patterns will also be detected and trigger an event to alarm all stakeholders. The setup will be tested in a real life environment.

1. Introduction

The ongoing aging in our modern society leads to the tendency to allow older persons to stay as long as possible in their home environment. Although they are mainly able to independently organize themselves, it is for a certain group nevertheless necessary to observe their activities of daily living (ADL). Examples of such activities include sleeping, cooking, making a phone call, visiting the toilet, washing, etc.

Our goal is to automatically detect changes in ADL patterns. These changes include both acute and gradual changes. Acute changes are abnormal events that are critical and require an immediate alarm. Examples of such abnormal events include: fall incidents, water or gas that keeps running or a sudden general absence of activity. On the other hand, we also want to detect gradual changes. These changes are important for an early detection of problems such as (early stage) dementia. Examples of such changes are sleeping disorders, ADL decline and behavioral disturbances. The information about these activities and changes in behavior can then be presented to the caregivers (including family members) to adapt older people's care plans, and as a consequence, increase their quality of care and quality of life. Hence, allowing them to stay longer at their homes.

Until now, research towards the automatic recognition of daily activities has been focused mainly on data from video [1] or from wearable wireless sensors such as accelerometers [2]. Many of these systems tend to be expensive and intrusive to the living environment. Also, some research shows that ADL can be classified by monitoring public utilities, but focus often on one sensor type like water consumption [3].

In this research, we aim to detect ADL in a non-intrusive way by fusing outputs of several sensor types in the house (Fig. 1). The setup will be installed and validated in a real life environment.

Appearing in Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands. Copyright 2011 by the author(s)/owner(s).

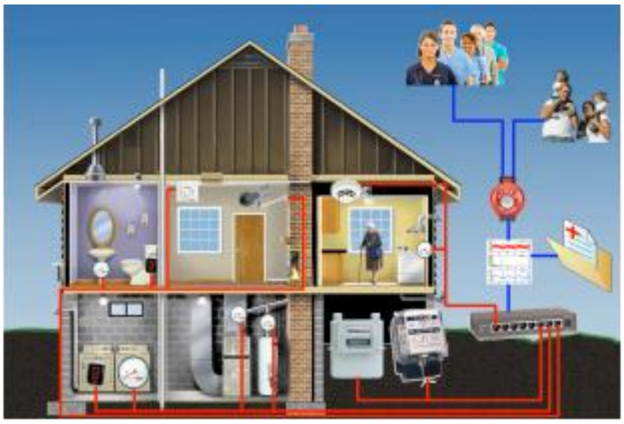


Fig 1. Sensors monitored

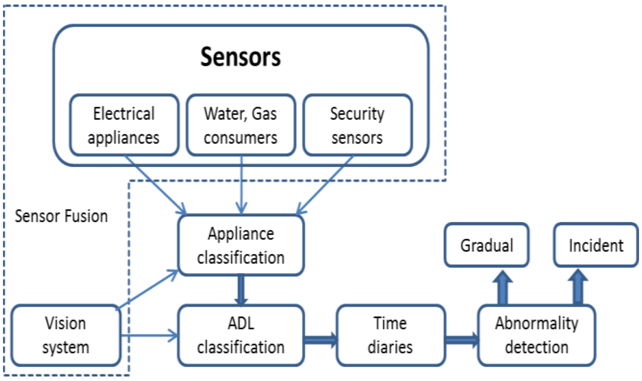


Fig 2. Overview of system

2. Methods

Fig.2 shows an overview of the method of underlying research.

Recognition of electrical appliances will be done by evaluating the total current and all relevant extracted features. Measuring of electricity, water and gas will be done at the central entrance point in the basement.

As for the vision software, we build further on the results of the FallCam project [4] which uses vision to automatically detect fall incidents.

To recognize ADL, we will use a combination of several detection mechanisms: the output of sensors measuring consumption of electricity, water and gas, together with security sensors and video cameras. This fusion of sensors, which is unique in this research domain, will provide a robust classification of ADL.

By fusing position information with appliance recognition information, it is easier to distinguish between appliances with the same electrical signature.

ADL classification based on detected appliances and output from the vision system (motion, position, posture detection) will be done using machine learning algorithms. By using the detected ADL, together with recorded time stamps, time diaries can be constructed from which a regular living pattern can be synthesized over a longer period of time.

Subsequently, we will also investigate to automatically detect changes in these activity patterns, again using machine learning techniques. These changes can either be incidental (e.g., person falling, leaving an appliance on, etc.) or long term trending, which could be an indication of health issues (e.g., cognitive decline). If such a detected abnormality occurs, an event alarm can be generated to the stakeholders (care taker, family, medical file, etc.)

Another important aspect of this project is that instead of only using lab data, we will do actual measurements in a real life setting: we deploy the setup in three living quarters of single living elderly for one year to calibrate our findings and use this data to develop a prototype.

3. Conclusion

In this paper we have described how we will implement a non-intrusive, contactless system for the detection and classification of ADL for the elderly people living alone. This enables us to generate regular patterns of living and detect abnormalities towards these patterns, both sudden and gradual. These abnormalities can trigger an alarm event to caretakers, family, medical file, etc. The results of this research will make it more feasible for single elderly people to sustain their independency longer and to provide a system to assist care takers in their support towards elderly.

Acknowledgments

This work is partly financed by EU EraSME (FP7), project Amacs (IWT 100404): Automatic Monitoring of Activities using Contactless Sensors.

References

[1] D. Andersen, R.H. Luke, J.M. Keller, M. Skubic, M.J. Rantz, M.A. Aud, “Modeling Human Activity From Voxel Person Using Fuzzy Logic”, IEEE Transactions on Fuzzy Systems, Vol. 17, No. 1, Feb. 2009.

[2] S. Im, I. Kim, S. Chul Ahn, H. Kim, “Automatic ADL classification using 3-axial accelerometers and RFID sensor”, Proc. IEEE Int. Conference on Multisensor Fusion and Integration for Intelligent Systems, Seoul, Korea, 2008

[3] J. Fogarty, C. Au, S.E. Hudson, “Sensing from the Basement: A Feasibility Study of Unobtrusive and Low-Cost Home Activity Recognition”, Proc. of the 19th Annual ACM Symposium on UIST’06, Montreux.

[4] G. Debard, J. Van Den Bergh, B. Bonroy, M. Deschodt, E. Dejaeger, K. Milisen, T. Goedemé, B. Vanrumste, “FallCam: Practical Considerations in Implementing a Camera-based Fall Detection System”, Positioning and Context-Awareness International Conference 2009.

Ensemble-Training: Ensemble Based Co-Training

Jafar Tanha, Maarten van Someren, and Hamideh Afsarmanesh
J.Tanha,M.W.vanSomeren,h.afsarmanesh@uva.nl
Informatics Institute, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands

Keywords: Semi-Supervised Learning, Co-Training, Ensemble Learning

1. Introduction

There are several different methods for semi-supervised learning. Here we consider co-training (Blum & Mitchell, 1998). Classifiers are trained using two views of data, usually subsets of features. Each classifier predicts labels for the unlabeled data including a degree of confidence. Unlabeled instances that are labeled with high confidence by one classifier are used as training data for the other. In this paper, we propose two improvements for co-training. First we consider co-training as an ensemble of N classifiers that are trained in parallel and second we derive a stop-criterion, using a theorem by Angluin and Laird (Angluin & Laird, 1988) that describes the effect of learning from uncertain data.

Two key issues in Co-Training are (1) measuring the confidence in labels that are predicted for the unlabeled data and (2) a criterion for stopping. Co-Training aims at adding a subset of the most confidently predicted labels. At some point labels will be noisy and cause the result of learning to become worse, a form of "overfitting". Problems (1) and (2) could be solved in an empirical way, using a holdout set of labeled data or some resampling scheme on the labeled dataset but Semi-Supervised Learning is used for learning tasks where labeled data is scarce. We use a theorem from PAC-learning (1) that relates the number of training data to the probability that a consistent hypothesis has an error larger than some threshold for a setting with training data with a certain error in the labels. We use an ensemble of learners for co-training and we use the agreement between the predictions of labels for the unlabeled data to obtain an estimate of the labeling error. Using this we can estimate the effect of learning on the error of the result of adding the new labeled data to the training set. In particular we use a theorem by Angluin and Laird (Angluin & Laird, 1988). If we draw a sequence σ of m data points then if

$$m \geq \frac{2}{\epsilon^2(1-2\eta)^2}\ln(\frac{2N}{\delta}) \tag{1}$$

where ϵ is the classification error of the worst remaining candidate hypothesis, η (< 0.5) is an upper bound on the classification noise rate, N is the number of hypothesis, and δ is the confidence, then a hypothesis H_i that minimizes disagreement with σ will have:

$$Pr[d(H_i, H^*) \geq \epsilon] \leq \delta \tag{2}$$

where $d(,)$ is the sum over the probability of elements from the symmetric difference between the two hypothesis sets H_i and H^*. Adding data with labels that have a probability of being incorrect means that m is increased and t hat η must be adjusted using the probability of an incorrect label for the current labeled set and of the newly labeled set. We fix δ and we assume that N is approximately constant. In that case we can calculate ϵ. The noise rate in this training set can be estimated by:

$$\eta_{i,j} = \frac{\eta_L|L| + \hat{e}_{i,j}|L_{i,j}|}{|L| + |L_{i,j}|} \tag{3}$$

From this we derive a criterion for whether adding data reduces the error of the result of learning or not. In Ensemble-training each component classifier h_j is first trained on the original labeled data. Next ensembles are built by using all classifiers except one. These ensembles predict the class of the unlabeled data and also the error rate is estimated from the agreement between the classifiers. After that a subset of U,unlabeled data, is selected by ensemble H_p for a classifier that will reduce its error. Here a threshold on the improvement can be used. This is then added to the labeled training data and the estimated error rate of these data is adjusted and used as training data for the "held-out" classifier. The selected unlabeled data for the subset in each training process is not removed from the unlabeled data U because may be the other component classifiers use them as well. This is repeated until no more data improve the error.

2. Evaluation and Conclusion

We compared our method to Tri-Training, Co-Forest (Li & Zhou, 2007) and Self-Training on a eight datasets from the UCI repository. Ensemble-training gave the best results on four of these. On average the accuracy was 1.63% higher than Co-Forest, which was second best. Though based on an approximation of the error rate, Ensemble-training gives good results compared to other methods.

References

Angluin, D., & Laird, P. (1988). Learning from noisy examples. *Mach. Learn.*, *2*, 343–370.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the eleventh annual conference on Computational learning theory* (pp. 92–100). New York, NY, USA: ACM.

Li, M., & Zhou, Z.-H. (2007). Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *37*, 1088 –1098.

Comparing Vessel Trajectories using Geographical Domain Knowledge and Alignments

Gerben K.D. de Vries G.K.D.deVries@uva.nl
Informatics Institute, University of Amsterdam, Sciencepark 904, 1098 XH, Amsterdam, the Netherlands

Willem Robert van Hage wrvhage@few.vu.nl
Computer Science, VU University Amsterdam, de Boelelaan 1081a, 1081 HV, Amsterdam, the Netherlands

Maarten van Someren M.W.vanSomeren@uva.nl
Informatics Institute, University of Amsterdam, Sciencepark 904, 1098 XH, Amsterdam, the Netherlands

1. Introduction

In this paper (de Vries et al., 2010) we present an alignment based similarity measure that combines low-level vessel trajectories with geographical domain knowledge, such as the name and type of the regions that vessels pass through and stop. We use this similarity measure in a clustering experiment to discover interesting behavior and in a classification task to predict the type of the vessel for a trajectory. The combination of information gives the best average classification accuracy. For both clustering and classification we use kernel based algorithms.

2. Trajectories & Geographical Domain Knowledge

We define a trajectory as $T = \langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle$, ignoring the temporal dimension. The number of vectors in T is denoted as: $|T|$. In the *stop* and *move* model of (Spaccapietra et al., 2008), the trajectories in our experiment are moves. They are delimited by the vessel entering the area of observation or starting, and the vessel leaving the area of observation or stopping.

The geographical domain knowledge comes as two simple ontologies. One, **A&C**, contains the definitions of different anchorages, clear ways, and other areas at sea. The other ontology, **H**, defines different types of harbors, such as liquid bulk and general cargo. For both ontologies, we created a SWI-Prolog webservice (van Hage et al., 2010) to enrich vessel trajectories with geographical features. The first service returns a set of specific type, label pairs corresponding to the regions in **A&C** that intersect with a given point. We create a sequence of sets of geo-labels $T^L = L_1, \ldots, L_{|T|}$ for a trajectory T with this service. For the start and end of a trajectory we define objects that contain information whether the vessel is stopped and if so at what harbor or region. We discover this harbor using the second webservice, which matches a point to the nearest harbor in **H** that is within range and returns the label and specific type of this harbor. If there is no harbor close, we use the first webservice.

3. Trajectory Similarity

For the sequences T and T^L we compute similarity using an edit distance alignment, which we discovered in previous work (de Vries & van Someren, 2010) to perform the best on a vessel trajectory clustering task. To compute an edit distance, we need a substitution function and a gap penalty. The substitution function for trajectories T is defined as: $\mathbf{sub}_{\text{traj}}(\langle x_i, y_i\rangle, \langle x_j, y_j\rangle) = -\|\langle x_i - x_j, y_i - y_j\rangle\|$, i.e. the negative of the Euclidean distance. We take the value for the gap penalty g from the mentioned previous work. For T^L, the substitution function $\mathbf{sub}_{\text{lab}}(L_i, L_j)$ expresses how many labels the sets of labels L_i and L_j have in common. We set g as the minimally possible $\mathbf{sub}_{\text{lab}}$ score.

The similarity $Sim(S, T)$ between two sequences S and T is the score of the alignment that has the maximum edit distance score for all possible alignments between these sequences, divided by $|S| + |T|$ to give the average score per element. In the experiments we use kernel based algorithms. For all sequences T_i and T_j in a set of sequences $\mathcal{T}$, we compute a kernel K as: $K(i, j) = Sim(T_i, T_j)$, then we normalize K and turn it into a kernel by $K = 1 - \frac{K}{\min(K)}$. For trajectories T we get a kernel K_{traj} and for sequences

Appearing in Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands. Copyright 2011 by the author(s)/owner(s).

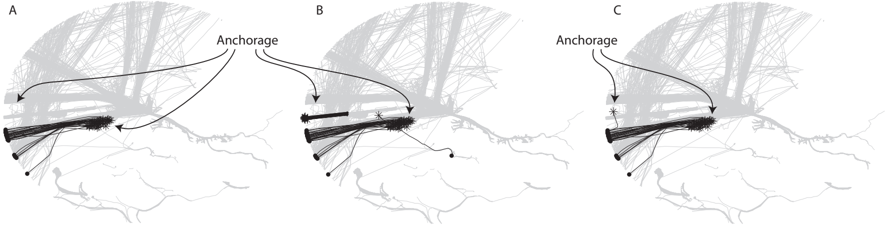


Figure 1. Example of a cluster of trajectories showing anchoring behavior. The example cluster is shown in black against the entire dataset in gray. The start of trajectory is indicated by a dot, the end by an asteriks.

of sets of geo-labels T^L we get a kernel K_{lab}. The similarity between two start/end objects can immediately be put into kernel form and is determined by whether the vessel is stopped or not and how much labels there are in common. This gives us a kernel K_{start} for the start objects and a kernel K_{end} for the end objects. $K_{\text{all}} = w_1 K_{\text{traj}} + w_2 K_{\text{lab}} + w_3 K_{\text{start}} + w_4 K_{\text{end}}$ combines the four kernels above. Clearly, this kernel is symmetric, but it is not guaranteed to be positive semi-definite.

4. Experiments

Our experimental dataset consists of 1917 vessel trajectories in a 50km radius area around the Port of Rotterdam, collected using the Automatic Identification System (AIS). The trajectories are compressed with the algorithm in (Gudmundsson et al., 2009), reducing the data by 95%, thus reducing computation time drastically. This compression improves performance on a vessel trajectory clustering task (de Vries & van Someren, 2010) using the same alignment.

For the clustering experiment we used weighted kernel k-means (Dhillon et al., 2007), with $k = 40$. We created kernels for 3 different weight settings of K_{all}: equal combination of domain knowledge and raw trajectories, K_{comb}, only raw trajectory information, K_{raw}, and only domain knowledge, K_{dom}. This results in a number of interesting clusters. In Figure 1A we see a cluster from clustering with K_{comb} that shows trajectories that enter the area from the west and anchor in one specific anchoring area. In B and C we plotted the most similar cluster from clustering with K_{raw} and K_{dom}, respectively. In Figure 1B there are also trajectories included that do not show the anchoring behavior, because we only consider raw trajectory information. We see the opposite in Figure 1C, where we have only anchoring behavior, but in different anchoring areas.

We also did a classification experiment, predicting the vessel's type. In total there are 18 types, available from AIS. For classification we used a support vector machine (SVM), with the same kernels as for clustering, in a 10-fold cross validation set-up. The classification accuracy for K_{all} was 75.4%, for K_{raw} 72.2%, and for K_{dom} 66.1%. All results differed significantly under a paired t-test with $p < 0.05$.

5. Conclusion & Future Work

The similarity measure that we defined was applied in a clustering task and we gave an example of discovered interesting vessel behavior that is a combination of both raw trajectories and geographical information. We also used the measure in classification to predict vessel types where the combined similarity showed the best performance in terms of classification accuracy. We plan to apply the measure in the task of outlier detection to discover strange vessel behavior.

References

de Vries, G., van Hage, W. R., & van Someren, M. (2010). Comparing vessel trajectories using geographical domain knowledge and alignments. *ICDM Workshops* (pp. 209–216). IEEE Computer Society.

de Vries, G., & van Someren, M. (2010). Clustering vessel trajectories with alignment kernels under trajectory compression. *ECML/PKDD (1)* (pp. 296–311). Springer.

Dhillon, I. S., Guan, Y., & Kulis, B. (2007). Weighted graph cuts without eigenvectors – a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*, 1944–1957.

Gudmundsson, J., Katajainen, J., Merrick, D., Ong, C., & Wolle, T. (2009). Compressing spatio-temporal trajectories. *Computational geometry*, *42*, 825–841.

Spaccapietra, S., Parent, C., Damiani, M. L., de Macêdo, J. A. F., Porto, F., & Vangenot, C. (2008). A conceptual view on trajectories. *Data & knowledge engineering*, *65*, 126–146.

van Hage, W. R., Wielemaker, J., & Schreiber, G. (2010). The space package: Tight integration between space and semantics. *T. GIS*, *14*, 131–146.

Scale-Independent Forecasting Performance Comparison

Wessel Luijben, Zoltán Szlávik, Daniel Dahlsveen **{W.LUIJBEN, Z.SZLAVIK, D.P.DAHLSVEEN}@VU.NL**
VU University Amsterdam, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands

Abstract

An evaluation methodology consists of three parts: a performance measure, a validation procedure and a significance test. We point out theoretical weaknesses in commonly used methodologies for comparing forecasting performance on time series data and introduce a new methodology that does not suffer from these weaknesses. An empirical evaluation is performed on both real world and synthetic datasets. Results show competent performance discrimination.

1. Introduction

Computer algorithms that create accurate forecasts about the future are of great value in many fields. A crucial step in the development of these algorithms is the evaluation of their performance. The comparison of the performance of multiple algorithms, amongst different datasets, is a challenging task. The M3 competition (Mandrakis & Hibon, 2000) is an example of a competition where algorithms are compared on multiple datasets. The M3 competition has prompted many discussions on how to interpret different performance measures, and how to assign winners to future competitions.

A big challenge is to find a performance measure which is independent of the scaling of the data. Most scaling invariant measures show undesired behavior in special cases, such as division by zero, or unsymmetrical loss for negative and positive errors. In addition, it is important to use a validation procedure which is in line with a typical forecasting task. A typical forecasting task has a fixed horizon, meaning that the distance in time between the prediction and last observed value is fixed. A procedure that meets this requirement is called Predictive Sequential Validation (PSV). PSV is outlined as follows: use all previously recorded data to train a model, wait *h* time steps and compare the predicted value (P) with the observed value (O), repeat previous step for *n* consecutive time points. In machine learning terms this corresponds to an ever growing training set and a test set containing only a single test point. Retraining a model every time when one wishes to make a prediction is uncommon in machine learning, and can be computationally expensive. Furthermore, the goal is not only to give an estimate of future performance, but also to give a confidence interval of future performance. This is important because a confidence interval, independent of the scaling of the data, makes it possible to compare results of different research. The t-test is widely used for this purpose, but a problem with time series data is that at least one major assumption of the t-test is violated: the assumption that errors are i.i.d.

In our work we address all three parts of a time series evaluation methodology: We present a novel measure, Relative Error Ratio (RER). In addition, we argue that PSV is advantageous over fixed split validation. Finally, we introduce a modification to the t-test which uses a different null hypothesis to incorporate the notion of time.

2. Notation & Equations

$$O\text{: Observed} \quad (1)$$

$$P\text{: Predicted} \quad (2)$$

$$Q\text{: Baseline} \quad (3)$$

$$MAPE\text{: } \left|1 - \frac{P}{O}\right| \quad (4)$$

$$sMAPE\text{: } \frac{|O-P|}{|O|+|P|} \quad (5)$$

$$A\text{: } \sum \mathrm{err}_O(P) \quad (6)$$

$$B\text{: } \sum \mathrm{err}_O(Q) \quad (7)$$

$\mathrm{err}_O(\cdot)$ is an arbitrary non-negative loss function

$$RER\text{: } \begin{cases} 1 - \frac{A}{B} & A<B \\ 0 & A=B \\ -1 + \frac{B}{A} & A>B \end{cases} \quad (8)$$

3. Commonly Used Measures & RER

Hyndman & Koehler (2005) state that commonly used measures for predictive performance are: Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Relative Error (MARE). Although proven optimal in some cases[†] MSE and MAE are not invariant to the scaling of the data and thus cannot be used to compare the performance amongst different datasets.

The measure MAPE (eq. 4) is often used in forecasting competitions. The problem with MAPE is that when O is close to zero, error scores tend to infinity. To overcome this problem Chen & Yang (2004) used a modified version of MAPE called sMAPE (eq. 5), which is undefined when both O and P are equal to zero. The biggest problem with sMAPE is that it puts extra weight on values that are close to zero, which is counterintuitive for most real-world tasks.

An alternative group of measures use a second predictor called a baseline predictor to make errors relative and comparable over different datasets. The proposed measure (eq. 8) falls into this category. RER is never undefined, and works in combination with MAE, MSE, or any other error measure that is non-negative. The baseline can be any predictor, for example: a simple baseline algorithm that always outputs the *last known target value* or a sophisticated algorithm implementing a *state of the art* forecasting technique.

Appearing in Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands. Copyright 2011 by the author(s)/owner(s).

[†] Chen & Yang (2004), show that the MSE is the optimal choice when errors have normal distributions and MAE is the optimal choice when errors have double exponential distributions.

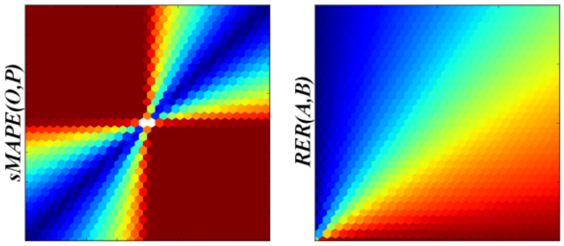


Figure 1 .The error landscape of sMAPE and RER. Inputs of RER are errors (only positive), whereas inputs to sMAPE are real values. Note that sMAPE is not defined at the origin, and is non-discriminative in both the top-left and bottom-right quadrant, whereas RER is defined everywhere, and is fully discriminative.

4. Significance Testing

Many papers skip statistical tests altogether and only report the mean and average error. Some authors even claim that statistical tests for time series analysis are unnecessary (Armstrong, 2007). This might be true for the analysis of one algorithm on one particular task, but as mentioned in the introduction, this severely limits the ability to compare results of different research. When people do perform a statistical test, in order to generate a confidence interval, they often use a t-test based on the mean, variance, and number of samples. Unfortunately, the independence assumption of test points, as used by the t-test, and many other statistical tests, clearly does not hold for time series data. Errors at time point t+1 are likely to be dependent on errors at time point t. The independence assumption basically eliminates the notion of time.

We propose a statistical test very similar to the t-test, but instead of using the entire dataset to calculate a single confidence interval, we calculate how the confidence interval evolves as a function of time. So for each time point we run a t-test taking into account all the data up to this point. Using this test it is possible to see important trends in the average error, and the variance in the error. As an example see Figure 2. Here we see the average RER score stabilize roughly at 0.5 and the bounds at 0.4 and 0.6. Since the RER score has stabilized it is very likely that P will also outperform Q on future data.

5. Predictive Sequential Validation

Single Split Validation can be outlined as follows: split the data into two parts. Use the first part for training and the second part for testing. Since the training set is fixed a model is trained only once. This is an advantage when training is slow. A disadvantage of single split validation is that the predictive horizon is not fixed, and therefore it is out of line with most real world tasks.

PSV, as noted in the introduction, is in line with most real world tasks. But as a disadvantage, it is computationally expensive in combination with a predictive algorithm that retrains its entire model when a new data point arrives, rather than updating its model. Examples of updatable algorithms are online linear regression (LR) and k-nearest neighbors (k-NN).

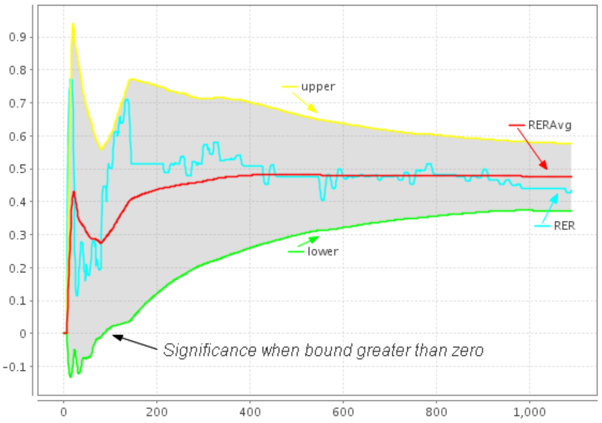


Figure 2. The RER confidence interval evolving as a function of time. The blue line shows the RER for each time point. The red line shows the average RER up to each time point. The yellow and green lines show the upper and lower bound. Observe that it takes roughly 100 test points before P is deemed significantly better than Q. This figure is generated on real world data from the weather domain comparing k-NN with the '*always output last known target value*' baseline.

6. Experiments & Results

As a proof of concept, we created three synthetic datasets using the following formulas: $f_1(t)$ = noise, $f_2(t)$ = t + noise, $f_3(t)$ = k-NN(t) + noise. Furthermore, we implemented linear regression forecasting and k-Nearest Neighbor forecasting. Note that LR is able to fit noiseless f_2 perfectly while k-NN is not; similarly k-NN is able to fit noiseless f_3 while LR is not. Running the algorithms on the datasets confirms our expectations. On f_1 LR and k-NN perform equally well, with RER scores around zero. On f_2 LR performs significantly better than k-NN, with the lower bound RER eventually greater than zero. On f_3 LR performs significantly worse than k-NN, with the upper bound RER eventually lower than zero. Furthermore we applied LR and k-NN to a weather dataset and a stock market dataset and sensible confidence intervals were observed.

7. Conclusion

Early results suggest that the proposed methodology is a good alternative to commonly used methodologies, but additional empirical research is required to validate this. For future work it will also be interesting to use this research to reanalyze the results of the M3 competition.

8. References

Armstrong, J. (2007). *Significance tests harm progress in forecasting*. International Journal of Forecasting, 23, 321-327.

Chen, Z. & Yang, Y. (2004). *Assessing Forecast Accuracy Measures*. [Online]. Available www.stat.iastate.edu/preprint/articles/2004-10.pdf

Hyndman, R. J. & Koehler B. (2006). *Another look at measures of forecast accuracy*. International Journal of Forecasting. 22, 679-688.

Makridakis, S. & Hibon, M. (2000). *The M3-competition: results, conclusions and implications*. International Journal of Forecasting, 16, 451–476.

Finding Fraud in Health Insurance Data with Two-Layer Outlier Detection Approach

R.M. Konijn W. Kowalczyk

Keywords: fraud detection, outlier detection, LOF-score

Abstract

Conventional techniques for detecting outliers address the problem of finding isolated observations that significantly differ from other observations that are stored in a database. For example, in the context of health insurance, one might be interested in finding unusual claims concerning prescribed medicines. Here, each claim may contain information on the prescribed drug (its code), volume (e.g., the number of pills and their weight), dosing and the price. Finding outliers in such data can be used for identifying fraud. However, when searching for fraud, it is more important to analyse data not on the level of single records, but on the level of single patients, pharmacies or GP's.

In this paper we present a novel approach for finding outliers in such hierarchical data. The novelty of our approach is to combine standard techniques for measuring outlierness of single records with conventional methods to aggregate these measurements, in order to detect outliers in entities that are higher in the hierarchy. We applied this method to a set of about 40 million records from a health insurance company to identify suspicious pharmacies.

1. Introduction

The inspiration for this paper comes from a real life fraud detection problem in health insurance, in the pharmacy domain. The goal of fraud detection in this context is to identify the most suspicious pharmacies that could possibly be involved in fraudulent activities, rather than identifying single claims that are suspicious. The main reason for not focusing on single outliers, is that recovering money from single claims is costly, and that it can harm the relationship between an insurance company and the involved pharmacy, especially in the case of false positives. On the other hand, if the insurance company can detect substantial fraud linked to multiple claims of the same pharmacy, this business relationship is no longer so important and a vigorous money recovery action can follow.

In contrast to typical approaches for finding single outliers, (Chandola et al., 2009), we propose a novel method for finding *groups* of outlying records that belong to the same class. Our method was successfully applied to a large set of health insurance claims, helping to identify several pharmacies involved in fraudulent behaviour.

2. Our Approach

Our method for detecting group outliers works in two stages. In the first stage we calculate outlier scores for single records. We use here classical methods for outlier detection that are based on distance measures, (Angiulli & Pizzuti, 2002), or density estimation, (Breunig et al., 2000).

Next, we calculate a statistic to measure the 'outlierness' of each groups of records, where groups form logical entities. In our case these entities are formed by all claims related to a pharmacy, or a combination of a pharmacy and a type of medication. We apply four different statistics that are used to define the final outlier score of these entities: (1) a rank based statistic, (2) a weighted rank based statistic, (3) a statistic based on the binomial distribution, and (4) a statistic that is based on the mean of the outlier score. These statistics can be applied in different situations for different outlier scores.

The statistics can be computed over different segments of the data to obtain the final score. Extra information about outlying entities is obtained by constructing

Appearing in *Proceedings of the 20^{th} Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the author(s)/owner(s).

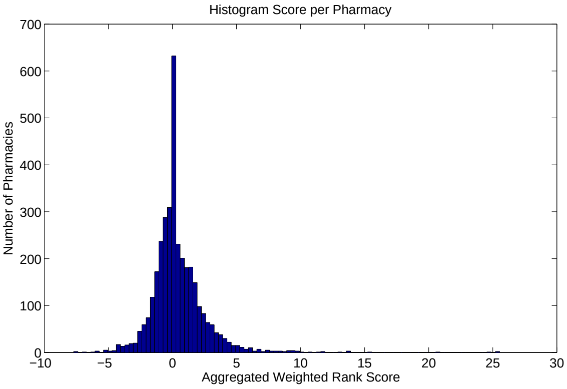


Figure 1. Histogram of the Weighted Rank Score Statistic. There are two observations with a score higher than 25, which are the most suspicious pharmacies.

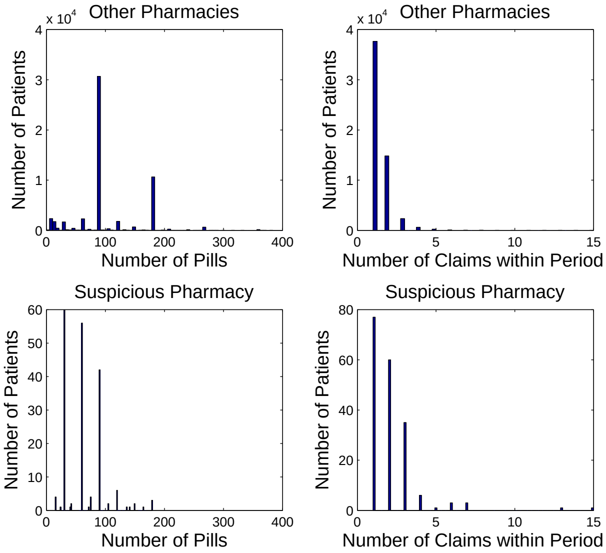


Figure 2. Histograms of the "number of pills prescribed" and the "number of claims" variables for the drug type *Aspirin*, both measured during the same period, over all pharmacies. The two histograms below show the distribution of the same variables calculated for the suspected pharmacy. From these graphs it can be concluded that these distributions are different. The number of pills is much lower on average, while the number of claims is higher. This is a clear indication of *unbundling* fraud.

so-called *fraud sets*: sets of suspicious claims from the same pharmacy. A fraud set is a set of outlying records that should be removed from the whole set in order to make it "normal" again. Another, very useful instrument for displaying fraud evidence is an *fraud plot*: a plot of fraud amount versus outlier score of all records in the fraud set. Here, the *fraud amount* is defined as the total amount of money that is involved in the observations that are in the fraud set. The fraud plot can be used by fraud investigators to decide whether they should investigate the most likely fraud cases, or to focus on cases that are less suspicious, but involve high amounts of money.

3. Results

We applied our method to detect fraud in a set of 40 million claims from various pharmacies. For three different types of problems, we first calculated a single point score and then we identified "outlying pharmacies" by using one of the four statistics mentioned earlier.

A common type of fraud in health insurance is *unbundling*: a practice of breaking what should be a single charge into many smaller charges. Two other common types of fraud are: delivering more units than stated on the prescription (and thus charging more money, and consequently increasing the profit), and charging money for drugs that have never been delivered. We identified these types of fraud simultaneously by mining "Local Outliers" at the patient level and then calculating an aggregated score per pharmacy by means of the weighted rank statistic. The most suspicious pharmacies (the ones with the highest score, see Figure 1) do indeed have strange claim behaviour.

We zoom in at the top outlier and plot the distributions of two variables: the "number of pills prescribed" and the "number of claims" against similar plots taken over all remaining pharmacies, see Figure 2. It is clear that the distributions of these variables are very different from the distributions of other pharmacies.

References

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. *Principles of Data Mining and Knowledge Discovery*, 43–78.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: identifying density-based local outliers. *SIGMOD Rec.*, *29*, 93–104.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, *41*, 15:1–15:58.

The Discovery of Criminal Behavior as a Ranking Problem

Keywords: forensic intelligence, class inbalance, SVM, ranking, information retrieval

Vaisha Bernard VAISHA@VAISHA.NL
Cor J. Veenman C.VEENMANL@NFI.MINJUS.NL
Netherlands Forensic Institute, Laan van Ypenburg 6, 2497 GB The Hague, The Netherlands

Abstract

When data mining for forensic investigations, we are typically confronted with strongly inbalanced classes. Moreover, the labels of the non-target or negative class are usually not confirmed. In other words, the non-positive objects are unlabeled. For these situations classification methods are not well suited. We propose to approach these problems as ranking problems. With a true investigation dataset, we show the improvement over the prior probabilities using the ranking approach.

1. Introduction

When data mining for forensic research, a typical situation is a data base with few recognized offenders and loads of unknowns. The unknowns are unlabeled objects that are for a greater part negatives, or non-offenders. The goal is to find among those unlabeled objects, likely offenders.

Unlabeled objects

This task can be considered a classification task, that is, separate the data set in offenders and non-offenders. However, to learn a classifier from a dataset, it should be a labeled data set. So the first problem is how to treat the unlabeled objects.

Strongly unbalanced classes

Secondly, in these forensic research scenarios, the number of target objects is orders of magnitude lower than the number of non-targets. Typically in the data base there are 102-103 targets and 105-106 non-targets. In other words the classes are strongly unbalanced.

Non-discriminating attributes

The third problem is that the attributes of the object records are usually not very discriminating. Any model learned for class separation will have modest performance. Accordingly, the number of false positives will be high. For two reasons such a situation is undesirable. First, with hardly any suspicion, it is legally and ethically not acceptable to check many people without success. That is, non-informative attributes implies that the suspicion will be low. Second, with limited resources it is even impossible to check people systematically. How can such a situation be improved? One option is to get more informative and discriminating attributes. This can for instance be achieved by combining several data bases or by combining other data sources like sensor data, surveillance camera videos, and photos taken by witnesses. For legal and ethical reasons this may, however, not be allowed. Even with additional data, classification performance can be too low.

2. Method

We propose to consider the problem of finding likely offenders among the database as a ranking problem. The objective is to yield a ranking of objects where the top-most objects are the most likely offenders. We learn the model based on the labeled offenders and consider the unlabeled objects as non-offenders. Because a fraction of the objects will be offenders, this assumption is for a greater part of the objects correct.

Recently, several ranking methods have been developed mainly aimed at information retrieval applications. All these methods are based on some form of support vector machine formulation. Methods exist for several types of ranking criterions, ranging from

Appearing in *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.* Copyright 2011 by the author(s)/owner(s).

Area under the ROC (AUC), to Precision at K and Mean Average Precision (MAP). In this study, we compaired these state-of-the-art methods to more traditional classification methods. For the classification methods, we used the posterior probability of the offender class as ranking criterion.

3. Datasets

The forensic dataset at hand consists of home addresses for which illegal irregular living situations need to be investigated. These situations often lead to problems with neighbors and the neighborhood in general. The municipality of concern wants to act more proactively by sampling more informed. The dataset consists of approximately 300,000 addresses from which 288 addresses have been established as positive in the sense of this investigation. In other words, one in thousand addresses has a known illegal irregular living situations. Moreover, for the other 300,000 addresses the situation is unknown.

For each a range of features is available, some are linked to the specific address and others are linked to the neighborhood in which the address resides. The latter type of features are therefore the same for all addresses in the neighborhood. Among the first class of features are: the number of households at the address, the number of persons enlisted who are over 18 years of age, the number of rooms, the total surface area, the period the house was built, the type of building, the type of municipal land use plan, the type of owner, and the level of administrative over-occupation. Since only the administrative truth is known, the features are probably not strongly correlated with illegal living situations.

4. Results

As performance criterion to compare the ranking methods, we used AUC, and Precision at 100. The last measure relates the most closely to the investigation practice, where, based on capacity, a limited sample is taken to put further research efforts on. The performance measures are estimated through a 10-fold stratified cross-validation procedure.

We applied only linear methods to this ranking problem. We used several traditional classification methods: Fisher Linear Discriminant (FLD), Logistic Regression (LogReg), and the Support Vector Machine (SVM). The methods that closely relate to the proposed performance measures are: SVM-Rank, SVM Precision@K (K=100), SVM-ROCArea, and SVM-MAP. For all SVM methods the C parameter is tuned also with a 10-fold stratified cross-validation procedure.

The ranking results are listed below:

Method	AUC	Precision@100
LogReg	0.907 ± 0.031	$\mathbf{0.035 \pm 0.017}$
FLD	0.927 ± 0.023	0.033 ± 0.017
SVM	0.925 ± 0.021	0.033 ± 0.019
SVM-Rank	0.929 ± 0.021	0.030 ± 0.016
SVM-Precision@100	0.914 ± 0.031	0.030 ± 0.018
SVM-ROCArea	$\mathbf{0.933 \pm 0.019}$	0.030 ± 0.017
SVM-MAP	0.920 ± 0.024	0.033 ± 0.019

Table 1. Performance of all ranking methods. The best scores per performance measure are printed in bold face.

5. Conclusion

In this study, we proposed to define forensic investigation problems as ranking problems. We demonstrated the ranking approach on a forensic dataset with strong class inbalance between the positive and negative class. Several specific ranking methods together with traditional classification methods were applied to the problem. In turned out that the prior probability of 0.1 positives in the top 100 ranked objects, could be improved to 3.5 objects in the top 100 on average (precision at 100 estimated using cross-validation). Moreover, the methods designed for ranking performance, such as SVM-Rank, SVM Precision@100, SVM-ROCArea, and SVM-MAP did not obtain the best performance scores for Precision at 100. Though, SVM-ROCArea indeed had the highest AUC score, for which it has been designed. The relative underperformance of the ranking methods for Precision at 100 could be explained by their sensitivity for label errors. That is, the labeling of the negatives in the training dataset is not confirmed. These negative labels are assumed, because a fraction of the objects will be positives. Further study is needed to explore this issue.

(Exceptional) Workflow Mining: A Stepwise Approach for Extracting Reliable Event Logs from Corporate Network Data

Guido Demmenie G.DEMMENIE@ROTTNIC.NL
Jan van den Berg J.VANDENBERG@TUDELFT.NL
Virginia Dignum M.V.DIGNUM@TUDELFT.NL
Jos Vrancken J.L.M.VRANCKEN@TUDELFT.NL
Delft University of Technology, Faculty of Technology, Policy and Management, Jaffalaan 5, 2628 BX Delft, NL

Menno Israël MENNO@HOLMES.NL
Netherlands Forensic Institute, Laan van Ypenburg 6, 2497 GB The Hague, NL

Keywords: Fraudulent Behavior, Forensics, Workflow Mining, Corporate Network Data, Event Logs

1. Introduction

Modern IT-environments that support large corporate organizations in their financial management suffer from many information security threats including those caused by people who execute ingenious fraudulent activities. In forensics, the challenge is to come up with evidence for such fraudulent behavior. Post-incident discovery of (potentially) criminal financial activities from corporate network data (defined as the complete set of files found on all storage devices of an organization) is a difficult task. Classical auditing and accountancy tools are not applicable since they are typically not suited for performing complex data analysis tasks.

The given discovery task is a typical problem to be solved by Machine Learning (ML) tools. Using such tools for finding evidence of performed criminal financial activities is not a new topic. Current literature on the topic (for references see (Demmenie, 2011)) reveals that (*a*) availability of a set of structured data is assumed to be true, and (*b*) the (deviating) patterns one is looking for are (more or less) known in advance. However, the evidence searching data miners might be less informed, e.g., in cases where it is known *that* certain fraudulent activities have been taken place but not the ways of how this is done. We further assume that the fraudulent activities result into certain anomalous patterns, i.e., patterns that deviate from the regular business activity patterns. This implies that we can discover fraudulent activities using a multi-step procedure by (1) extracting, for each business task, the regular patterns, and (2) detecting the anomalous patterns.

Thinking about regular, task-related patterns in a business environment, workflow patterns are a natural choice. Literature again shows quite some examples. In addition, the related algorithms also assume availability of structured data, here in terms of reliable *event logs* describing the different, time-stamped (trans)actions related to the (sub-)activities of a business task. This makes the *extraction of event logs a fundamental first step* for finding different business activity patterns.

2. Event Log Extraction, a Stepwise Approach

We assume having available a large set of CND containing all kinds of data around financial transactions. We also assume that the data contain (usually unique, consecutive) invoice numbers related to financial transactions. Then, the proposed stepwise approach for event log extraction can be executed as follows:

(1) *File Indexing:* File indexing is needed to reduce search times related to searching for specific data in next sub-steps.

(2) *Invoice Number Mining:* Finding invoice numbers is needed to detect identifiers, each one of which relates to a single financial transaction.

(3) *Transaction Related Document Collection:* Document collection is about finding all the documents that contain information about unique transactions.

(4) *Sub-activity Extraction:* Based on the documents collected, the related sub-activities per transaction can be extracted, e.g., based on the names of the docu-

ments and the time stamps used.

(5) *Event Log Extraction:* Based on the identified subactivities and the order of their execution, full event logs of the financial transactions can be induced.

Knowing the event logs, the natural next step is to induce the *regular* financial activity patterns around financial transactions. Examples of those patterns are regular workflows. Event log-based workflow mining tools are fortunately already available (W. van der Aalst et al., 2004). To test our approach, we executed a case study.

3. Regular Workflow Mining, an example

The case study made use of a (confidential) real-world data set originating from an international company containing 300 gigabyte of data in total (Demmenie, 2011). The suspicion existed that certain financial irregularities had taken place.

Event Log Extraction. Event log extraction was executed first according the stepwise approach proposed. For indexing, an existing tool (W. Alink et al., 2006) was used resulting into 9.6 million quickly searchable files. To mine invoice numbers, a specific filter was used. Some additional tricks were needed to extract the 'true' invoice numbers from all numbers found. The document collection step (where all documents containing invoice numbers were learned) resulted into a set of 618 documents. Next, it was possible to detect a set of financial transactions-related activities where context information as present in the files and knowledge from experts in the field had to be used. Finally, the event logs have been created consisting of a list of 6367 transactions related to 194524 sub-activities over a time-frame of just over a year.

Regular Workflow Learning. Having available the event logs, an existing workflow mining tool entitled ProM (W. van der Aalst et al., 2004) has been used to create the graph of regular workflows. The result of this task is shown in figure 1.

The resulting model is quite complex with many connections, loops and shortcuts. Using the tool we created a simulation of the mined model. This simulation visualizes all the cases in the event log and shows how they travel through the model. We observed some interesting behavior. It appears for instance that when an invoice comes into the activity 'outstanding', (the activity in which all outstanding invoices are collected), it often loops back to that same activity for a considerable amount of times. A deeper inspection is needed to discover exceptional patterns.

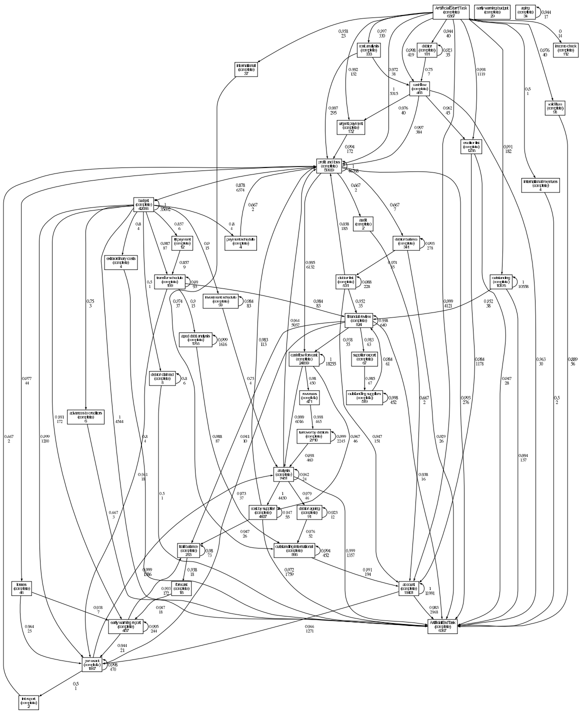

ArtificialStartTask (complete) 6367
early warning budget (complete) 29
aging (complete) 34
international (complete) 37
cost analysis (complete) 333
debtor (complete) 181
income check (complete) 112
cashflow (complete) 465
void item (complete) 98
urgent payment (complete) 172
creditor list (complete) 1235
profit and loss (complete) 55609
international revenues (complete) 4
budget (complete) 42066
audit (complete) 2
till payment (complete) 12
payment schedule (complete) 4
debtor balance (complete) 341
extraordinary costs (complete) 4
transfer schedule (complete) 199
debtor list (complete) 531
outstanding (complete) 10875
investment schedule (complete) 99
financial review (complete) 824
aged debt analysis (complete) 1755
cashflow forecast (complete) 24899
supplier export (complete) 67
debtor claimed (complete) 8
revenues (complete) 471
outstanding suppliers (complete) 519
advances to creditors (complete) 6
turnover by debtors (complete) 2710
analysis (complete) 7451
cost by supplier (complete) 4607
debtor ageing (complete) 91
trail balance (complete) 283
outstanding international (complete) 886
losses (complete) 46
forecast (complete) 18
account (complete) 18401
early warning report (complete) 457
ArtificialEndTask (complete) 6367
per exact (complete) 1867
int report (complete) 2

Figure 1. Resulting workflow model from induced event logs.

4. Conclusions

A first result has been obtained in the development of a methodology for the post-incident discovery of ingenious fraudulent activities. It concerns a stepwise approach for the extraction of a set of reliable event logs from a huge set of unstructured CND. Continued research is expected to result into a robust forensic methodology that can help forensic workers to effectively and efficiently collect required evidence.

References

Demmenie, G. (2011). Workflow mining, a stepwise approach for extracting event logs from corporate network data. Master's thesis, Faculuty of Technology, Policy and Management, Delft University of Technology.

W. Alink, R. Bhoedjang, P. Boncz, & A. de Vries (2006). Xiraf - xml-based indexing and querying for digital forensics. *Digital Investigation, Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)* (pp. 50–58).

W. van der Aalst, T. Weijters, & L. Maruster (2004). Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, *16*, 1128–1142.