Benelearn 2017: Proceedings of the Twenty-Sixth Benelux Conference on Machine Learning, Technische Universiteit Eindhoven, 9-10 June 2017

Benelearn 2017: Proceedings of the Twenty-Sixth Benelux Conference on Machine Learning

Editors: Wouter Duivesteijn, Mykola Pechenizkiy, George Fletcher, Vlado Menkovski, Eric Postma, Joaquin Vanschoren, and Peter van der Putten

Welcome!

Benelearn is the annual machine learning conference of the Benelux. It serves as a forum for researchers to exchange ideas, present recent work, and foster collaboration in the broad field of Machine Learning and its applications. These are the proceedings of the 26^{th} edition, Benelearn 2017.

Benelearn 2017 takes place largely on the campus of the Technische Universiteit Eindhoven, De Zaale, Eindhoven. The Friday programme is located in De Zwarte Doos (see `https://goo.gl/maps/XgKEo7JxyTC2`), and the Saturday programme in Auditorium (see `https://goo.gl/maps/B3PnpuCjgMJ2`). The conference dinner on Friday evening is the only off-campus event; this takes place in the DAF Museum, Tongelresestraat 27, 5613 DA Eindhoven (see `https://goo.gl/maps/zNLrhpSqimk`).

As part of the main conference programme, we organize three special tracks: one on Complex Networks, one on Deep Learning, and one Industry Track. Distributed over all tracks, contributing researchers not only span all three Benelux countries, but also include affiliations from ten additional countries.

We thank all members of all programme committees for their service, and all authors of all papers for their contributions!

Kind regards,
The Benelearn 2017 organizers

Organization

Conference Chairs: Wouter Duivesteijn, Mykola Pechenizkiy

Complex Networks Track Chair: George Fletcher

Deep Learning Track Chairs: Vlado Menkovski, Eric Postma

Industry Track Chairs: Joaquin Vanschoren, Peter van der Putten

Local Organization: Riet van Buul

Programme Committee Conference Track

Hendrik Blockeel, K.U. Leuven
Sander Bohte, CWI
Gianluca Bontempi, Université Libre de Bruxelles
Walter Daelemans, University of Antwerp
Tijl De Bie, Ghent University, Data Science Lab
Kurt Driessens, Maastricht University
Ad Feelders, Universiteit Utrecht
Benoît Frénay, Université de Namur
Pierre Geurts, University of Liège
Bernard Gosselin, University of Mons
Tom Heskes, Radboud University Nijmegen
John Lee, Université catholique de Louvain
Jan Lemeire, Vrije Universiteit Brussel
Tom Lenaerts, Université Libre de Bruxelles
Marco Loog, Delft University of Technology
Martijn van Otterlo, Vrije Universiteit Amsterdam
Yvan Saeys, Ghent University
Johan Suykens, KU Leuven, ESAT-STADIUS
Celine Vens, KU Leuven Kulak
Willem Waegeman, Ghent University
Marco Wiering, University of Groningen
Jef Wijsen, University of Mons
Menno van Zaanen, Tilburg University

Programme Committee Complex Networks Track

Dick Epema, Delft University of Technology
Alexandru Iosup, Vrije Universiteit Amsterdam and TU Delft
Nelly Litvak, University of Twente
Taro Takaguchi, National Institute of Information and Communications Technology
Yinghui Wu, University of California Santa Barbara
Nikolay Yakovets, Eindhoven University of Technology

Programme Committee Deep Learning Track

Bart Bakker, Philips Research
Binyam Gebre, Philips
Ulf Grossekathofer, Holst Centre and IMEC
Mike Holenderski, TU Eindhoven
Dimitrios Mavroeidis, Philips Research
Decebal Constantin Mocanu, TU Eindhoven
Elena Mocanu, TU Eindhoven
Stojan Trajanovski, Philips Research

Programme Committee Industry Track

Hendrik Blockeel, K.U. Leuven
Kurt Driessens, Maastricht University
Murat Eken, Microsoft
M. Israel, Erasmus MC
Arno Knobbe, Leiden University
Arne Koopman, ASML
Hugo Koopmans, DIKW Consulting
Wannes Meert, KU Leuven
Dejan Radosavljevik, T-Mobile Netherlands
Ivar Siccama, Pega
Johan Suykens, KU Leuven, ESAT-STADIUS
Jan Van Haaren, KU Leuven
Cor Veenman, Netherlands Forensic Institute and Leiden University
Mathias Verbeke, Sirris
Lukas Vermeer, Booking.com
Willem Waegeman, Ghent University
Jef Wijsen, University of Mons
Jakub Zavrel, TextKernel
Michiel van Wezel, Dudok Wonen

Contents

Conference Track

Invited Talks

Data mining, social networks and ethical implications

Toon Calders TOON.CALDERS@UANTWERP.BE
Universiteit Antwerpen

Abstract

Recently we have seen a remarkable increase of awareness of the value of data. Whereas companies and governments mainly used to gather data about their clients just to support their operations, nowadays they are actively exploring new applications. For instance, a telecom operator may use call data not only to bill its customers, but also to derive social relations between its customers which may help to improve churn models, and governments use mobility data to chart mobility patterns that help to assess the impact of planned infrastructure works. I will give an overview of my research in this fascinating area, including pattern mining, the analysis of influence propagation in social networks, and ethical challenges such as models that discriminate.

Appearing in *Proceedings of Benelearn 2017*.

Generalizing Convolutions for Deep Learning

Max Welling M.WELLING@UVA.NL
Universiteit van Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

Abstract

Arguably, most excitement about deep learning revolves around the performance of convolutional neural networks and their ability to automatically extract useful features from signals. In this talk I will present work from AMLAB where we generalize these convolutions. First we study convolutions on graphs and propose a simple new method to learn embeddings of graphs which are subsequently used for semi-supervised learning and link prediction. We discuss applications to recommender systems and knowledge graphs. Second we propose a new type of convolution on regular grids based on group transformations. This generalizes normal convolutions based on translations to larger groups including the rotation group. Both methods often result in significant improvements relative to the current state of the art.

Joint work with Thomas Kipf, Rianne van den Berg and Taco Cohen.

Appearing in *Proceedings of Benelearn 2017*.

Dynamics and mining on large networks

Jean-Charles Delvenne JEAN-CHARLES.DELVENNE@UCLOUVAIN.BE
Université catholique de Louvain

Abstract

A network, i.e. the data of nodes connected by edges, often comes as the support of dynamical interactions. For example a social network is often measured as the trace of an information flow (phone calls, messages), energy and phase information flow through power networks, biochemical networks are the skeleton of complex reaction systems, etc. It is therefore natural to mine network-shaped data jointly with a real or modelled dynamics taking place on it. In this talk we review how dynamics can provide efficient and accurate methods for community detection, classification, centrality and assortativity measures.

Appearing in *Proceedings of Benelearn 2017*.

The transformative impact of automated algorithm design: ML, AutoML and beyond

Holger Hoos H.H.HOOS@LIACS.LEIDENUNIV.NL
Universiteit Leiden

Abstract

Techniques from artificial intelligence — and especially, machine learning — are fundamentally changing the way we solve challenging computational problems, and recently, automated machine learning (AutoML) has begun to take this to a new level. In this talk, I will share my perspective on the success of ML and AutoML, and discuss how the fundamental concepts and tools that enable both have a much broader impact than commonly perceived. In particular, I will highlight the role of a fruitful interplay between machine learning and optimisation in this context, comment on general approaches to automated algorithm design, and share my thoughts on the next big challenge.

Appearing in *Proceedings of Benelearn 2017*.

Conference Track

Research Papers

Extracting relevant discussion from Reddit Science Science AMAs

L.F.J.M. Kanters WIEKE.KANTERS@STUDENT.RU.NL
Radboud University, Nijmegen, the Netherlands

Keywords: text mining, spam detection, reddit, naive Bayes

Abstract

The social network and content aggregation website Reddit occasionally hosts Q&A sessions with scientists called science AMA (Ask Me Anything). These science AMAs are conducted through the comment system of Reddit which has a tree structure, mark-up and community driven feedback on both users and comments in the form of "karma" scores.

Most of the actual discussion in these science AMAs tends to be of high quality. However a large number of the comments are superfluous and not really part of the conversation with the scientist. The goal of this project is to determine if text mining methods can be used to filter out the unwanted comments. A secondary goal is to determine the relative importance of Reddit meta-data (tree structure, karma scores, etc) compared to the actual content of the comments.

The python Reddit API was used to retrieve the AMAs. The CoreNLP tools were used to extract tokens, sentences, named entities and sentiment. These were combined with other information, like Reddit meta-data and WordNet, and used to extract features. The classification was done by a Gaussing naive Bayes classifier using the scikit-learn toolbox.

Classification using all features or only text-based features was effective both yielding a precision/recall/f1-score of 0.84/0.99/0.91. Only using Reddit based features was slightly less effective, yielding 0.89/0.63/0.74. Only using a single WordNet based similarity feature still worked, yielding 0.81/0.99/0.89.

Appearing in *Proceedings of Benelearn 2017*.

1. Introduction

On reddit there is the tradition of the Ask Me Anything, or *AMA*, threads. These are a kind of informal interview or online Q&A session with whomever started the thread (the OP or original poster), anybody can participate and ask questions. For about 3 years the /r/science subreddit, a subforum dedicated to science, has been doing AMAs with scientists as a kind of science outreach. As a result there are now well over 600 different online AMAs with scientists covering a wide variety of subjects with more being done each week. The strict moderation in /r/science has resulted in a subreddit culture that tends towards serious discussion which, combined with the enthusiasm of the scientists involved, yields AMAs of an exceptionally high quality. The informal nature of reddit allows lay-people easy access, while also allowing for more in-depth questions. The hierarchical structure of the comment section as well as the lack of time constraints nature of an AMA, a particular enthusiastic OP might still be answering questions days later, encourages follow-up discussion. And the /r/science community has a decent number of scientists among its members, who are recognizable due to *flair* next to their username, so experts other than the OP are likely to join in the discussion. However despite this, large parts of these AMAs are still superfluous, consisting of unanswered questions, tangential discussions, there is clearly a lot of knowledge to be found in these AMAs but some manner of filtering might be required first.

In order to archive these AMAs, and to assign them their doi-number so they can actually be referenced in scientific literature, the Winnower has been copying parts of these AMAs to their own website[1]. Some of the larger AMAs can end up having many thousands of comments, with only a tiny fraction of them actually being worth archiving. So the Winnower's applies a filter to these AMAs but it is rather crude. They take every comment that is at the 2nd level of the

[1] `https://www.thewinnower.com/topics/science-ama`

comment tree and is made by the scientist being interviewed, which tend to be answers to questions, and their parent comment, which would contain the question. Everything else, including follow-up discussion, is discarded.

The primary research question of this paper is *to what extent it is possible, using text-mining, to distinguish between informative & relevant comments and the rest.* A secondary research question is *to what extent reddit meta-data and comment hierarchy is necessary for this classification.*

2. Background

2.1. Reddit

Reddit (Reddit, 2016) is a social aggregation website where users can submit content (either links to a webpage or a piece of text), rate this content, comment on it, and of course, consume it. It is quite a large site, for example in 2015[2] alone it received 73.15 billion submissions and 73.15 billion comments written by 8.7 million users.

The *frontpage* of reddit, the entry point to the website, consists of a list of *submission* titles ordered by popularity. Popularity of a submission is determined by user feedback in the form of *upvotes* and *downvotes* fed into an unknown stochastic function yielding a *score*. Official reddit etiquette[3] states that one should vote based on whether or not "you think something contributes to the conversation", though in practice voting is also based on agreement or amusement. Further user feedback is possible by "gilding" which costs the gilder $4, confers benefits to the poster of the gilded content and places a little golden star medallion next to the submission. Each piece of content is submitted to a specific subreddit which functions both as a category and community of sorts, a user can subscribe to subreddits and their personal frontpage is a composite of the subreddits they are subscribed to. Usually when mentioning a subreddit the name is preceded by '/r/' because Reddit automatically turns such a mention into a link to the subreddit.

Each submission has an associated *comment section* where users can have a conversation. The conversation in a reddit comment section is tree, each comment being either a direct reply to the original submission or to another comment in the thread. Just as a submission the comments themselves can also be voted upon, or be gilded, and will usually be displayed in order of popularity. The user who started the thread, by submitting the piece of content the thread pertains to, is referred to as *OP*, short for original poster.

All the upvotes and downvotes for every submission and comment of a user are combined into a link and comment *karma*, by subtracting the sum of downvotes from the sum of upvotes. Each user's karma is publicly visible and tends to be used, in combination with length of time that user has been a *redditor*, as an informal indication of a user's reliability.

Moderation of reddit is generally handled by volunteer moderators, or *mods*, with responsibilities and permissions limited to the subreddits they moderate. Moderation policy varies from subreddit to subreddit.

Among the tools for mods are deletion of content, adding flair, banning and shadow-banning. *Flair* is a short, 64 characters, piece of text that can be used to customize submissions and users. *User flair* can be set by the user or a mod (depending on subreddit policy) and will be shown next to the user's username on submissions and comments. *Submission flair* can be set by the user who submitted the content or a mod (again depending on subreddit policy) and will be shown next to the submission title.

In /r/science submission flair is used to categorize submissions by field, which is fairly typical use for submission flairs in reddit. User flair policy in /r/science is quite unique, the mods use user flair to indicate the academic qualifications of a user (for example it might say "BS | Artificial Intelligence"), these qualifications are verified by the mods. The /r/science mods call this their "science verified user program"[4] the intention of which is to allow readers to distinguish between "educated opinions and random comments", verified users are also kept to a higher standard of conduct.

2.2. Related work

Weimer et al (Weimer et al., 2007) did work on automatically assessing post quality in online forums, they attempt to assess the usefulness of different types of features some of which are based on forum metadata. This is rather similar to the work in this paper. Their result was that classification based on sets lacking any forum based features performs slightly worse than classification based on sets including those features. Their work uses annotation based on user feedback through built-in features of the forum software and the general goal underlying the classification differs a bit from what is being done here.

[2] `https://redditblog.com/2015/12/31/reddit-in-2015/`

[3] `https://www.reddit.com/wiki/reddiquette`

[4] `https://www.reddit.com/r/science/wiki/flair`

Siersdorfer et al (Siersdorfer et al., 2010) did a similar study based on youtube comments. It may be worth noting that this study is from before youtube switched to google+ comments, when it was still feasible to moderate comment sections. The interesting thing here is that they found a significant correlation between the sentiment of a comment, as analyzed by SentiWordNet, and the scores users attributed to comments. Though again, just as with Weimer et all, the point of the classification is a bit different from what is being done here.

On a slightly different note Androutsopoulos (Androutsopoulos et al., 2000) et al compares the performance of a naive Bayes classifier on spam detection. Their pipeline is fairly simple, the most complex configuration used in their works merely employs a lemmatizer and stop-list, though despite this it manages to get good recall and precision on their email corpus. The use of a naive Bayes classifier is especially interesting since its transparent decision making process would allow one to easily assess the impact of each feature.

3. Methods

3.1. Data

The data was taken from the following two reddit AMAs:

- Hi reddit! Im Alice Jones, an expert on antisocial behaviour and psychopathy at Goldsmiths, University of London. I research emotion processing and empathy, focusing on childhood development and education. AMA![5] with 239 comments (172 used).
- Hi, my name is Paul Helquist, Professor and Associate Chair of Chemistry & Biochemistry, at the University of Notre Dame. Ask me anything about organic synthesis and my career.[6] with 234 comments (121 used).

The data was annotated manually based on whether or not a given comment was informative & relevant and would therefore be worth keeping, as if the annotator was editing down an interview. The information available to the annotator was the text of the comments themselves as well as the comment hierarchy. Normally in reddit sibling comments would be ordered by popularity but during annotation this ordering was randomized. Comments without any replies were not shown during annotation and were assumed to be not worth keeping, since they cannot be part of any discussion or question/answer pairs.

The data was annotated by two different annotators who each annotated all comments presented to them. The Cohen's Kappa is $\kappa = 0.45$, indicating a mere moderate agreement. In the interest of preserving as many relevant comments as possible a comment will be considered worth keeping if at least one of the annotators thought it was worth keeping.

The AMA by Alice Jones was used as the training set and the AMA by Paul Helquist as the test set.

3.2. Pipeline

3.2.1. Data gathering

The data was gathered using the Python Reddit API (Praw-dev, 2016) which allows one to essentially do and see from a Python script everything one would be able to do or read using the reddit website. This was used to gather the following from the AMAs:

- The text of the original submission, which contains an introduction of both the OP and the topic of the AMA.
- The text of each comment in the comment section.
- For the original submission and comment:
  - The amount of times it was gilded.
  - The karma (upvotes minus downvotes) it received.
  - The flair of the user who wrote it.
- For each of the users:
  - Whether the user currently has gold.
  - The total amount of comment karma for all their comments all over reddit.
  - The total amount of link karma they received.
  - Whether the user is a mod of any subreddit.
  - Whether the user was shadowbanned.
  - A breakdown of link and comment karma by subreddit it was gained on.

Note that the text retrieved was encoded in utf-8, contains xml character entity references and has markup in the markdown format.

All the data was stored in XML files while maintaining the hierarchy of the comment section.

3.2.2. Preprocessing and normalization

Preprocessing and normalization was done using the Stanford CoreNLP (Manning et al., 2014) language

[5] https://www.reddit.com/r/science/comments/4twmrc/science_ama_series_hi_reddit_im_alice_jones_an/

[6] https://www.reddit.com/r/science/comments/52k2gt/american_chemical_society_ama_hi_my_name_is_paul/

processing toolkit, the following built-in annotators were used:

- Tokenization
- Sentence splitting
- Lemmatization
- Named entity recognition
- Syntactic analysis
- Part of speech tagging
- Sentiment analysis

Any token was dropped if it was an url, if it is a stopword, or if it is neither a verb, noun, adjective or adverb. Though some information, like the number of urls in the comment, were kept as a feature.

3.2.3. Feature extraction

The features were extracted using a Python script creating a series of feature vectors for each comment, see the section 3.3 for details on the features. Most of the features simply consist of a piece of reddit meta-data or some quantity derived directly from the text. However two of the features (t_{ws} and c_{fws}) make use of WordNet (Fellbaum, 1998) implementation of the Natural Language Toolkit (Bird et al., 2009) based on the lemma and part-of-speech information determined by CoreNLP. Another three features (c_+, $_c$-, c_o)make use of the SentiWordNet (Baccianella et al., 2010) toolkit which expands the WordNet with negative, objective and positive sentiment values for words. The spellings based features c_m and c_{co} are based on the enchant python library[7]. The curseword feature c_{cu} is based on the profanity python library[8].

3.2.4. Classification and Evaluation

Both Classification and evaluation of the classification based on the extracted feature vectors was done using the scikitlearn toolbox (Pedregosa et al., 2011). During classification and evaluation one AMA was used as a test set, while the other was used as a training set, see section 3.1. The classifier used was a Gaussian Naive Bayes (Bishop, 2006) classifier because of its transparent nature. This transparency enabled a closer examination of the features as described in section 3.4.

3.3. Features

Ultimately all features of a comment are combined into one large feature vector prior to being used for classification. The features used were split into two categories:

- Features that depend on reddit meta-data and comment structure shown in table 2.
- Features purely based on the text of the comment, independent of reddit meta-data. Besides the ones shown in table 1 these also include token document frequencies.

The feature vector consists of the features shown in tables 2 and 1 followed by the document frequencies of a number of tokens. Which document frequency would be included was determined by taking the top N tokens, of the entire training set, ordered by document frequency. How many tokens were included, or hyperparameter N, is shown in section 4.1.

3.3.1. Similarity features

Two different types of similarity features were used, features that indicate how similar two comments are to one another.

One based solely on which exact tokens occurred in both comments, and how frequently they occurred. This similarity was defined as follow, where $df_{t,x}$ is the document frequency of token t in comment x:

$$\text{similarity}(x, y) = \sum_{t \in x,y} df_{t,x} df_{t,y} \tag{1}$$

The other similarity feature was based on WordNet path similarity. WordNet can determine a similarity between two tokens by measuring the distance between two tokens within the WordNet network, this is the path similarity ($\text{sim}(a, b)$). The WordNet similarity of two comments was determined by taking the average of the path similarity all possible combinations of tokens in both comments:

$$\text{WordNetsimilarity}(x, y) = \frac{1}{|x||y|} \sum_{a \in \text{x}} \sum_{b \in \text{y}} \text{sim}(a, b) \tag{2}$$

These similarities were used as elements in the feature vector. The similarity between a comment and the introductory comment made by the scientist was included. As well as the similarity between the user flair (the scientific credentials as shown on /r/science) of the commenter and the scientist doing the AMA.

3.4. Gaussian naive Bayes

Since the naive Bayes classifier is rather transparent, it is possible to look at the way a particular feature

[7] `https://github.com/rfk/pyenchant/`
[8] `https://github.com/ben174/profanity`

Table 1. Features independent of reddit meta-data and comment hierarchy. The last column, JS-divergence, shows an indication of how much that feature influences classification, see section 4.4 for details.

Feature	Description	JS-divergence
$\frac{c_p}{t_n}$	The number of paragraphs divided by the number of tokens.	$3.12 \cdot 10^{1}$
$\frac{c_{url}}{t_n}$	The number of hyperlinks divided by the number of tokens.	9.44
$\frac{t_{ne}}{t_n}$	The number of named entities divided by the number of tokens in the comment.	$1.33 \cdot 10^{-1}$
$\frac{t_c}{t_n}$	The number of correctly words divided by the number of tokens in the comment.	9.31
$c_?$	The number of sentences ending in a question mark in the comment.	$6.66 \cdot 10^{-1}$
c_m	The number of misspelled words in the comment.	$5.60 \cdot 10^{-1}$
c_p	The number of paragraphs in the comment.	$5.74 \cdot 10^{-1}$
c_+	The average positivity of the words in the comment according to SentiWordNet.	2.46
c_-	The average negativity of the words in the comment according to SentiWordNet.	1.56
c_{co}	Fraction of correctly spelled words in the comment.	9.31
c_{cp}	If the user uses capitals and periods at the start and end of their sentences 1, otherwise 0.	$4.79 \cdot 10^{2}$
c_{lc}	The number of full-caps words of more than 3 character in the comment.	$1.25 \cdot 10^{-1}$
c_o	The average objectivity of the words in the comment according to SentiWordNet.	$1.95 \cdot 10^{1}$
c_{sen}	The average sentiment according to Stanford NLP by sentence in the comment.	2.10
c_{url}	The number of hyperlinks in the comment.	2.02
t_n	The number of tokens in a comment.	$5.88 \cdot 10^{-1}$
t_s	The similarity between this comment and the initial comment made by OP.	1.48
t_{cu}	The number of cursewords in the comment.	1.30
t_{ne}	The number of named entities found by Standford NLP.	1.02
t_{ws}	The WordNet similarity between this comment and the initial comment made by OP.	$7.70 \cdot 10^{2}$

Table 2. Features dependent on reddit meta-data and comment hierarchy. The last column, JS-divergence, shows an indication of how much that feature influences classification, see section 4.4 for details.

Feature	Description	JS-divergence
c_a	The number of ancestral comments of the comment in the tree.	$9.80 \cdot 10^{-2}$
c_c	The number of child comments of the comment in the tree.	1.90
c_g	If the comment has been gilded 1 otherwise 0.	0.00
c_k	The log amount of karma of the comment.	1.62
c_s	The number of sibling comments of the comment in the tree.	$9.29 \cdot 10^{-1}$
c_{opa}	If a comment made by OP is among the ancestral comments 1 otherwise 0.	0.00
c_{opc}	If OP replied to this comment 1 otherwise 0.	1.08
c_{opd}	If a comment made by OP is among descendant comments 1 otherwise 0.	1.20
c_{opp}	If the parent comment was made by OP 1 otherwise 0.	$2.05 \cdot 10^{-2}$
u_b	If the user was shadowbanned 1 otherwise 0.	0.00
u_f	If the user has /r/science flair, otherwise 0.	1.04
u_g	If the user has gold 1 otherwise 0.	$1.52 \cdot 10^{-1}$
u_m	If the user is a mod of any subreddit 1 otherwise 0.	1.06
u_d	If user was deleted 1 otherwise 0.	$8.81 \cdot 10^{1}$
u_{fs}	The similarity between comment's user flair and the flair of OP.	$2.19 \cdot 10^{1}$
u_{fws}	Wordnet similarity between comment's user flair and the flair of OP.	$2.19 \cdot 10^{1}$
u_{kc}	The log amount of comment karma of the user of the comment.	3.66
u_{kl}	The log amount of link karma of the user of the comment.	$9.12 \cdot 10^{-2}$
u_{op}	If the comment's user is also the OP 1 otherwise 0.	$1.31 \cdot 10^{1}$

impacts the resulting classification from a mathematical perspective. This would be a secondary method for finding the relative importance of features besides comparing classification performance.

Consider the following probabilistic model, which is used by Naive Bayes for classification, where all distributions p are Gaussian:

$$p(C_k|x_1,\ldots,x_n) = p(C_k)\prod_{i=1}^{n} p(x_i|C_k)$$

The predicted class will be the one with the highest probability according to the model. So say only feature x_j is being changed and one only has two classes, all the other features can simply be dropped:

$$p(C_1)\prod_{i=1}^{n} p(x_i|C_1) < p(C_2)\prod_{i=1}^{n} p(x_i|C_2)$$
$$p(C_1)p(x_j|C_1) < p(C_2)p(x_j|C_2)$$

And unless the difference in prior probability is quite high the difference between probability distributions $p(x_j|C_1)$ and $p(x_j|C_2)$ should determine which class is predicted. If these distributions were similar the value of x_j would influence the posterior probability only little, since either distribution would assign it a similar probability. Which means that the difference between $p(x_j|C_1)$ and $p(x_j|C_2)$ could be seen as a measure for the importance of feature x_j.

4. Results

4.1. Hyper parameter N

The optimal number N token document frequencies to include in the feature vector was determined as follows.

Figure 1 shows the recall, precision and f1 scores of the classification by N, where the document frequency of the top N tokens by document frequency were include in the feature vector, as well as the proportion of unique sets of document frequencies. This figure shows a slight trade-off between precision and recall. The document frequency used to order the tokens by document frequency is based on the comments in the training set only. For the purpose of these tests a random 20% of the training set was used as a hold-out test set.

An N of 575 was settled upon in order to maximize recall without needlessly increasing the number of document frequencies included in the feature vector. The preference for recall over precision corresponds to a preference for keeping interesting comments over discarding uninteresting ones.

4.2. All features

In order to determine if this classification works at all a test run was performed where every feature was used and the document frequency of the top $N = 575$ tokens.

All features

Truth \ Prediction	Discard	Keep
Discard	13	17
Keep	1	90
Precision	0.84	
Recall	0.99	
F1	0.91	

4.3. Reddit vs Text features

In order to determine if the use of reddit metadata based features has any effect the test has been repeated twice. Once with features solely derived from the comment text and the document frequency of the top $N = 575$ tokens:

Text features only

Truth \ Prediction	Discard	Keep
Discard	13	17
Keep	1	90
Precision	0.84	
Recall	0.99	
F1	0.91	

And once with on features solely derived form reddit meta-data, no text based features were used:

Reddit features only

Truth \ Prediction	Discard	Keep
Discard	23	7
Keep	34	57
Precision	0.89	
Recall	0.63	
F1	0.74	

4.4. Features

As discussed in section 3.4 the difference between the probability distributions underlying the Gaussian naive Bayes classifier could be seen as an indication of feature importance. The specific difference measure being used here is Jensen-Shannon divergence (a symmetric version of KullbackLeibler divergence). This

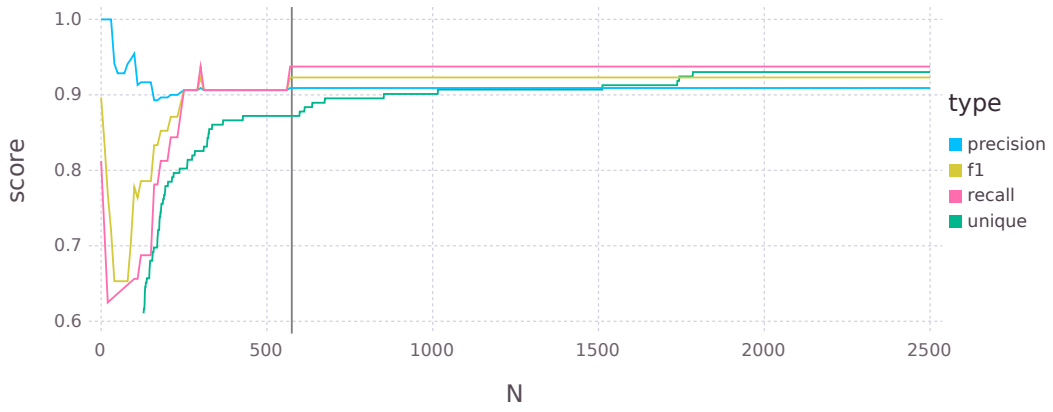


Figure 1. Performance of classification and proportion of unique bags in the dataset by N, the number of tokens whose document frequencies were included in the feature vector. The vertical line indicates the N used for further tests.

divergence basically shows how different the prototypical "keep" comment is compared to the prototypical "discard" comment based on a specific feature. Tables 2 and 1 show this divergence.

Because the WordNet similarity feature t_{ws} seems rather influential it might be interesting to see how it would perform on its own, even discarding the document frequencies ($N = 0$).

Wordnet Similarity t_{ws} only

		Prediction	
		Discard	Keep
Truth	Discard	9	21
	Keep	1	90
Precision		0.81	
Recall		0.99	
F1		0.89	

5. Discussion

Regarding the primary research question of whether or not it is possible, using text-mining, to distinguish between informative & relevant comments and the rest, it certainly seems possible. As shown in section **??** each of the classification methods, all features, reddit only features and text only features, have at least decent precision, f1 and recall scores. Though the performance of reddit only features is not as good as the rest.

Regarding the secondary research question whether or not reddit meta-data and comment hierarchy is important to this classification, the answer seems to be no. As shown in section **??** the difference between all features and text only features classification is nonexistent, while there does exist a difference between reddit only and text only feature classification. Perhaps most interesting is that the WordNet similarity feature on its own performs nearly as well as all features combined.

The other way of figuring out the relative importance of the features, suggested in section 3.4, would be to look at the inner working of the fitted Gaussian naive Bayes classifier. Consider tables 2 and 1.

On the reddit-dependent side the flair similarity features u_{fs} and u_{fws} seems to be of import, probably because it ends up identifying the scientist doing the AMA himself. The hierarchy related features c_{opc}, c_{opd} and c_c are important as well because they indicate that a comment was an integral part of the discussion. Admittedly this the interestingness of these features is dampened somewhat knowing that they do not really contribute to the classification. In hindsight features c_g and u_b were useless because no comment was gilded nor user shadowbanned in our data.

On the text-only side the WordNet based similarity t_{ws} really stands out, even more so than the normal similarity t_w, probably because comments that are semantically similar to the introductory text are likely to be relevant to the discussion. In section 4.4 it's even shown to be good enough to function its own. It is rather nice to see that the extra effort of including semantics pays off. One of the other WordNet, well SentiWordNet, based features c_o also seems influential, it is supposed to indicate the objectivity of

a comment a quality one would expect from scientific explanations. Other indicators of well formatted and well sourced comments, features c_{cp}, $\frac{c_p}{t_n}$ and $\frac{c_{url}}{t_n}$ seem also to be of import.

6. Conclusion

So it appears that it is quite possible to use text-mining to distinguish between informative & relevant comments and the rest. And that while reddit meta-data is quite useful it is not at all necessary for classification. To the point where even a single text-based feature, the WordNet similarity measure, performs better than all the reddit meta-data features combined.

The one real issue with these conclusions is that the amount of data used is quite small, mostly because annotating the data manually is quite time consuming. Initially using reddit comment karma as annotation was considered but the distribution of said karma is horribly skewed which led to issues. The vast majority of comments will never have had any feedback on them, they would have had no upvotes or downvotes, resulting in a karma of 1.

Also a lot more data was gathered than was actually used for this paper. Not just in raw quantity (228 different AMAs were automatically scraped from reddit) but also in terms of quality. Neither the breakdown of karma by subreddit or the markdown formatting was used. And the first would probably reveal a lot about the user, as a sort of fingerprint of their interests on reddit.

It might also be interesting to do this feature analysis using a classifier that does not make the independence assumption naive Bayes does.

Or it may be interesting to look into the usefulness of different semantics based features, like word2vec or any of the other WordNet based distance measures. Seeing as the one real stand-out is the WordNet based similarity measure. Which used on its own yields a performance nearly as good as all features combined. And could be interpreted as a kind of "on topic"-ness feature.

References

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D. (2000). An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 24–28.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). SentiWordNet 3 . 0 : An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining SentiWordNet. *Analysis*, *0*, 1–12.

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*, vol. 43.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. springer.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*, vol. 71.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 55–60).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Praw-dev (2016). PRAW: The Python Reddit API Wrapper.

Reddit (2016). Reddit FAQ.

Siersdorfer, S., Chelaru, S., & Augusta, V. (2010). How useful are your comments?: analyzing and predicting youtube comments and comment ratings. *Proceedings of the 19th international conference on World wide web*, *15*, 891–900.

Weimer, M., Gurevych, I., & Mühlhäuser, M. (2007). Automatically assessing the post quality in online discussions on software. *Proceedings of the ACL*, 125–128.

Locally versus Globally Trained Word Embeddings for Automatic Thesaurus Construction in the Legal Domain

I.G. Veul IVAN.VEUL@GMAIL.COM
Institute for Computing and Information Sciences, Radboud University, the Netherlands

Keywords: word embeddings, word2vec, global, local, thesaurus construction

Abstract

In this paper two different word embedding methods are explored for the automatic construction of a thesaurus for legal texts. A word embedding maps every word to a relatively low dimensional vector, which is then used to compare similarities between words. We use Word2Vec for the word embedding, which is an unsupervised learning method that defines a word based on its context. Words with similar contexts will then be considered similar. The unsupervised nature of Word2Vec allows for the construction of the thesaurus without requiring relevance feedback. A downside with the standard Word2Vec approach though is that the resulting word embeddings tend to be too general, when trained on an entire corpus. This paper studies whether training the word embeddings separately for different jurisdictions results in a better thesaurus. The thesauri are trained on the text of 300 000 Dutch legal rulings. To assess the performance of the globally and locally trained thesauri, they are compared to a manually constructed thesaurus, which is already being used for query expansion in the legal domain. The results show that there is a significant difference between the global and local thesauri, but that the global thesaurus actually outperforms the local thesaurus.

1. Introduction

Over the last 15 years the legal community has shifted from working with paper to working digitally, spawning specialized digital collections containing many legal documents. But with more and more information becoming available and the size of these collections growing rapidly, it has become increasingly difficult for legal experts to actually find relevant information in these collections. Documents might use synonyms or describe similar concepts in different terms. A legal expert who is not aware of these variations in the terminology can have a difficult time coming up with the right words to describe their information need and as a result might miss out on relevant documents.

This problem is amplified by the specialized nature of the documents (IJzereef et al., 2005): First of all, experts of different legal fields might use different terminologies. This means that an expert would require a detailed understanding of the field and the words used in order to effectively search for documents. Specialized documents also tend to contain abbreviations and acronyms, increasing their ambiguity. Finally, the vocabulary used in legal documents also varies over time, as new concepts and laws are introduced, making it difficult for experts to keep up with the terminology.

To combat this ambiguity problem, queries can be expanded with words that are closely related to the original words. This concept of query expansion has been an active research area since the 70s (e.g. Minker et al., 1972) and over the years a wide array of techniques has been tested (see related works for examples). One such technique is using words from a thesaurus to expand the query. A thesaurus is a collection in which words are grouped with other words that have a similar meaning or are otherwise related. Thesauri generally contain three types of word relations: synonyms, hierarchical and related (Hersh et al., 2000). Words that are synonyms can be used interchangeably and share the same meaning; words belonging to the hierarchical category share a broader/narrower relation; and the related category contains all other types of relationships between words that are considered important, for example two words being each others opposites.

Appearing in *Proceedings of Benelearn 2017*.

A downside of thesauri for query expansion is that their creation and maintenance takes a lot of time, is labor intensive and is prone to human error (Lauser et al., 2008). An alternative is to automatically construct a thesaurus, by extracting word similarities directly from the documents in the collection. A common approach for this is to use word embeddings. A word embedding is a mapping of a word to a low dimensional vector of real numbers. These embeddings can be used to calculate the distance between two words (for example using cosine similarity), which serves as a quantitative representation of the similarity between the words (Roy et al., 2016).

The interest in word embeddings has been refueled by the introduction of a new word embedding technique by Mikolov et al. (2013), called Word2Vec. Word2Vec uses a neural network to calculate the vector representations of words, by predicting the surrounding words of a given word. The advantages of Word2Vec are that it is easily accessible through the Word2Vec software package[1] and less computationally expensive than other word embedding techniques (such as Latent Dirichlet Allocation), which can get computationally very expensive with large data sets (Mikolov et al., 2013).

Word2Vec struggles though with generalization (Diaz et al., 2016), because the word embeddings are trained on the whole vocabulary. This effect could be augmented in collections with legal documents, since different fields of law require different interpretations and as a result might use words differently. The goal of this paper is to study if this effect can be mitigated by training Word2Vec separately for each legal field. This is done in a two stage process: Firstly this study aims to confirm that a thesaurus trained on the entire collection differs from a thesaurus trained on separate legal fields. Then this paper tries to answer whether the locally trained Word2Vec embeddings create a better thesaurus than globally trained embeddings, in the context of legal documents. The contribution of this paper is limited to the detection of related words and does not address the assignment of thesaurus relationships to these words.

2. Related Work

Over the years a broad range of possible query expansion techniques have been studied. Ranging from adding clustered terms (Minker et al., 1972) to state-of-the-art methods that use relevance models, such as RM3 (Abdul-Jaleel et al., 2004). The usage of thesauri for query expansion has shown mixed results, which is highlighted by Bhogal et al. (2007) in their review paper. Hersh et al. (2000) saw for example a general decline in performance, in their study assessing query expansion using the UMLS Metathesaurus, while in some specific cases their query expansion method actually showed improvement. IJzereef et al. (2005) observed consistent significant improvement when applying thesaurus query expansion to biomedical retrieval. A unique aspect about their approach was that they tried to take into account how terms from a thesaurus could benefit retrieval, by reverse engineering the role a thesaurus can play to improve retrieval. Tudhope et al. (2006) took a less technical approach and looked at possible difficulties for users, when using thesauri for query expansion.

Studies related to word embeddings for query expansion can be divided into two categories. The first category are studies in which word embedding techniques were directly used for query expansion. Roy et al. (2016) for example used similar terms based on Word2Vec embeddings directly for query expansion. Although their study showed increased performance on general purpose search tasks, the method failed to achieve a comparable performance with state-of-the-art query expansion methods. Diaz et al. (2016) used Word2Vec in a similar way, but showed the importance of locally training Word2Vec on relevant documents to overcome the generalization problem. In their study locally training the word embeddings significantly improved the performance for query expansion tasks, compared to globally trained embeddings.

The second category uses word embeddings to automatically construct thesauri. Navigli and Ponzetto (2012) for example used a combination of word embeddings from WordNet and Wikipedia to construct a cross-lingual, general purpose thesaurus. Claveau and Kijak (2016b) also used WordNet (in combination with Moby) to construct a thesaurus, but used a different approach to find related terms for the thesaurus. Instead of using cosine similarity measures directly to link terms to relevant terms, they formed documents from clusters of similar terms based on their word embeddings. Building the thesaurus was then done by finding the most relevant document for every term.

3. Method

This section describes the preprocessing of the data and the construction of the globally and locally trained thesauri from the data. A visual summary of this process is given in Figure 1.

[1] `https://code.google.com/archive/p/word2vec/`

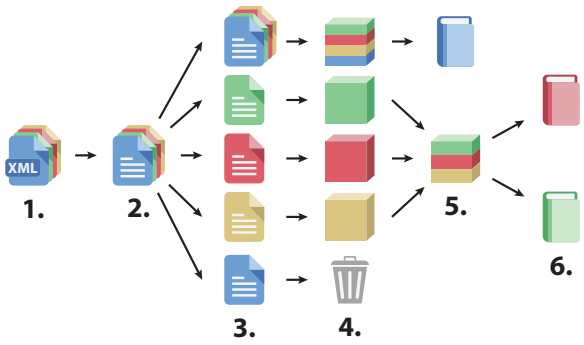


Figure 1. A visual summary of the method used to construct the thesauri. **1.** The process starts with 300 000 XML files of Dutch legal rulings. **2.** Sentences are extracted from the XML files. **3.** The sentence data is split into 5 groups: administrative law, civil law, criminal law, international public law and a group with all of the sentences. **4.** Word2Vec models are constructed for every group of the data, except for international public law which did not have enough data. **5.** A thesaurus is created from the global model, by taking the top ten most similar terms for each term in vocabulary. The three locally trained models are joined together. **6.** Two local thesauri are constructed from the joined local model. One that solves duplicate conflicts by taking the maximum similarity score and one that uses the average score.

3.1. Data

The data used to train the Word2Vec embeddings consisted of three hundred thousand court rulings, which were provided by Legal Intelligence[2]. The rulings were crawled from the Dutch governmental website that publishes court verdicts[3] and were represented as semi-structured XML files. Each file contained, among other things, the ruling of the court, a short summary of the case (*inhoudsindicatie*) and the jurisdiction to which the ruling belonged. The arrest and the summary were used as the text sources for the training of the word embeddings, whereas the jurisdiction was used to group the rulings for local training. The rulings belonged to one of four jurisdictions: administrative law, civil law, international public law or criminal law.

3.2. Reference Thesaurus

To evaluate the learned thesauri, the thesauri were compared to a ground truth in the form of a reference thesaurus. The reference thesaurus was the 2015 version of the *justitiethesaurus*[4]. The justitiethesaurus is a publicly available legal thesaurus aimed at query expansion for expert search in the legal domain (van Netburg & van der Weijde, 2015). It is created and maintained by the *Wetenschappelijk Onderzoek- en Documentatiecentrum* (WODC). The thesaurus was based on ten sources, which consisted of legal keyword lists and dictionaries, books on legal and immigration concepts, a book about country names and a book about the construction of thesauri (van Netburg & van der Weijde, 2015).
The reference thesaurus consisted of 5558 terms and each term had one or more related terms. In total there were 13 606 related terms (5877 unique). The justitiethesaurus used five types of relations to connect these terms, which are explained in Table 1. The thesaurus contained both single word terms as well as terms in the form of multi-word phrases. For example, '*aandeelhouders*' (shareholders) had '*algemene vergadering van aandeelhouders*' (general meeting of shareholders) as a related term.

Table 2. The number of sentences for each of the jurisdictions after preprocessing.

Jurisdiction	Sentence Counts
Total	40 523 303
Administrative Law	19 651 290
Civil Law	12 643 019
Criminal Law	8 223 217
International Public Law	5 777

3.3. Preprocessing

Before the text was used to train the Word2Vec embeddings, the text was split up into sentences using the sentence tokenizer from the nltk module for Python. For each sentence, all digits and punctuation symbols (except the '-') were removed, converted to lower case and then split on whitespaces. The number of sentences for each jurisdiction, excluding empty ones, are shown in Table 2. The sentences belonging to the international public law jurisdiction were discarded for local training, because there were not enough sentences to effectively train the word embeddings, but were included in the global model.

[2] `https://www.legalintelligence.com`
[3] `https://www.rechtspraak.nl/`
[4] `https://data.overheid.nl/OpenDataSets/justitiethesaurus\_2015.xml`

Table 1. Description of the five types of relations of the *justitiethesaurus* and their number of occurrences.

Relation	Description	Count
Narrow-term	Term lower in the hierarchy	1822
Broader-term	Term higher in the hierarchy	1822
Related-term	A term that is related, but not hierarchically	6458
Use	Reference to the preferred almost) synonymous term	1752
Used-for	Reference to an (almost) synonymous term. Used if the original term is the preferred term	1752

3.4. Training

Since a significant number of terms in the reference thesaurus were phrases, the `phrases` model[5] of the *gensim* module was used on all of the sentences in the collection, to learn common bigram and trigram phrases. The phrases were trained using the default settings and only taking into account phrases that occurred at least five times.

The Word2Vec embeddings were then trained on unigrams and the previously identified phrases, using the skip-gram implementation of gensim's Word2Vec model[6]. Training was done locally for the three remaining jurisdictions and globally on the entire text collection. The neural network consisted of 100 hidden nodes and used a maximum distance of five words. In other words, the context of a word was defined as the five words before and the five words after it. All terms that did not occur at least ten times were discarded.

After training the models, only the terms that occurred in both the reference thesaurus as well as the trained models were selected. This was required in order to compare the thesauri with the reference thesaurus. The term counts of the four models, before and after reduction, are shown in Table 3. Not all terms of the reference thesaurus were retrieved by the models. Many of the terms in the reference thesaurus that were not identified by the Word2Vec models, were phrases of two or more words (despite extracting commonly used phrases from the texts). This means that these words and phrases did not actually occur (often enough) in the training data.

Finally, the thesauri were constructed from the Word2Vec models by taking the ten most similar terms for each term, based on the cosine similarity of the embeddings.

[5] `https://radimrehurek.com/gensim/models/phrases.html`

[6] `https://radimrehurek.com/gensim/models/word2vec.html`

Table 3. The vocabulary size (the term count before reduction) and the term count in the reference thesaurus for all four models.

Model Name	Vocabulary Size	Count in Reference
Global	559 032	3585
Administrative Law	302 444	3147
Civil Law	267 466	2989
Criminal Law	168 664	2627

3.5. Combining Jurisdiction Thesauri

The final step before comparing global and local thesauri, was to combine the models for each jurisdiction into a single local thesaurus. This was done by concatenating the most similar terms, for each term in the jurisdiction models, and then ordering the similar terms based on their cosine similarity scores. If the same term showed up as a similar term in multiple jurisdiction models, the conflict was solved using two methods: The *maximum* method used the highest similarity score as the score for the combined thesaurus. The second method used the *average* similarity score as the new score.

After combining the jurisdiction models, the local thesauri covered 3526 terms, compared to the 3585 terms covered by the global thesaurus. This discrepancy was due to terms being discarded in the jurisdiction models for not occurring more than ten times, while they did occur more than ten times globally. The terms that were only present in the global thesaurus were ignored for the comparison between the thesauri.

4. Results

4.1. Similarity of Thesauri

Before analyzing the performance of the two types of thesauri, it was important to determine whether the globally and locally trained thesauri actually differed

significantly. For the comparison, the trained thesauri were considered as a list of rankings of related terms: one ranking for each term in the thesaurus. Comparing the global and local thesauri then meant that all these individual rankings had to be compared. This was done using the rank correlation coefficients Kendall's τ and Spearman's ρ. Figure 2 shows the histograms of the correlation coefficients for both types of local thesauri, when compared to the globally trained thesaurus. The blue bars denote the cases where the p-value of the correlation were insignificant with a significance level of $\alpha = 0.05$, whereas the green bars denote the significant correlations.

The histograms show that for both ranked correlation measures the coefficients are approximately normally distributed, where the majority of the coefficients are close to zero. This means that for most of the terms in the thesauri, there was little dependence between the rankings of the global and local thesauri. Moreover, the large majority of the rankings had insignificant correlation coefficients, meaning that the rankings were not significantly dependent. The correlation coefficients and their p-values thus show that it is unlikely that the globally and locally trained thesauri were similar.

4.2. Performance of Thesauri

To test whether the differences between the local and global thesauri were actually an improvement, the three thesauri were compared to a ground truth thesaurus. The comparison was done by treating the related terms of the ground truth thesaurus as relevant terms, that had to be retrieved by the rankings of most similar terms in the trained thesauri. This was done for each term in the ground truth thesaurus, which was then summarized for the entire thesaurus using r-precision and MAP at different values of k. The results are shown in Table 4.

The results show that the globally trained thesaurus performed consistently better than both the locally trained thesauri. Of the local thesauri, the thesaurus which used the average concatenation method performed the worst across the board. For all three thesauri, taking into account only the single most similar term (i.e. $k = 1$) resulted in the highest MAP scores. In other words, taking into account more terms did not improve the recall enough to compensate for the decrease in precision.

In general though, the MAP@k and r-precision scores are very poor. This means that a lot of irrelevant terms were retrieved by the trained thesauri, and thus none of the thesauri constructed from Word2Vec embeddings were very good.

Table 4. Comparison of the performances of the global and local thesauri, as evaluated on the ground truth thesaurus. The r-precision was calculated on the entire top ten similar terms for each term in the thesaurus.

Thesaurus	MAP@k			R-Precision
	$k = 1$	$k = 5$	$k = 10$	
Global	0.072	0.064	0.065	0.055
Maximum	0.066	0.058	0.060	0.048
Average	0.060	0.051	0.054	0.044

The quality of the trained thesauri does not just depend on how many terms are retrieved correctly, but also on the types of correctly retrieved terms. These are shown in Table 5. The results show no significant differences between the global and local thesauri, but there are differences within the term types. Most notably, it follows from the table that approximately half of the relevant terms, that were retrieved by the trained thesauri, were synonyms (terms with the 'use' and 'used-for' relations). These two types of relations though only accounted for approximately one-fourth of the number of related terms in the ground truth thesaurus (see Table 1). The trained thesauri thus had a bias towards synonyms, compared to the ground truth. On the other hand, terms with the 'related-term' relationship were underrepresented, only accounting for $24\% - 30\%$ of the terms instead of the 47% in Table 1. These two differences are most prevalent when only looking at the single most similar term in the trained thesaurus ($k = 1$). As k grows, and more terms with lower similarity scores are included, the relative number of synonyms retrieved decreases slightly as the relative number of terms with the 'related-term' increases slightly. This enforces the suspicion that the trained thesauri tend to assign higher similarity scores to synonyms than other types of related terms.

5. Discussion

5.1. Direct versus Indirect Evaluation

The performances of the trained thesauri were evaluated by comparing them to a ground truth thesaurus. This evaluation method is based on the assumption that the ground truth thesaurus is a reflection of what a good thesaurus should look like. Although the thesaurus used for this experiment was constructed by experts and has been improved upon for more than twenty years (van Netburg & van der Weijde, 2015), that does not necessarily mean that other good thesauri have to look similar. As a result, it is difficult to infer the quality of the thesauri solely based on the

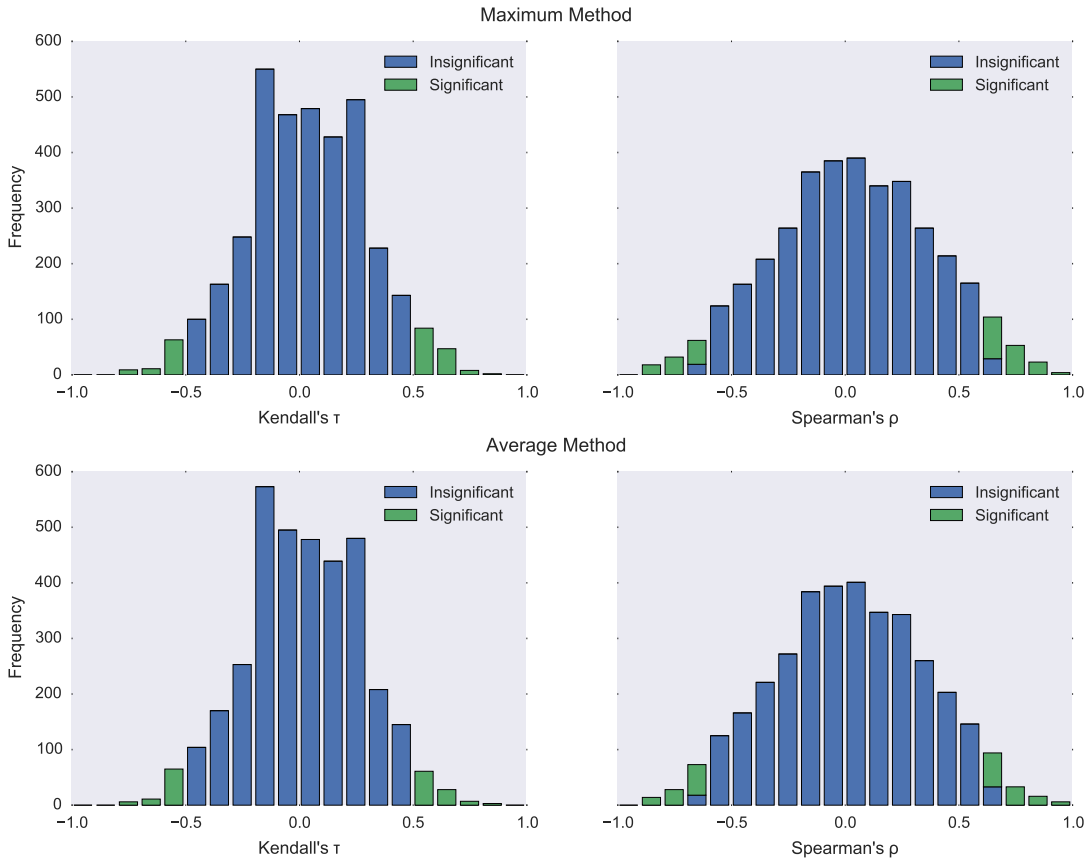


Figure 2. Similarities between the global thesaurus and both types of local thesauri expressed in the Kendall's τ and Spearman's ρ. The correlation coefficients are binned in twenty bins of size 0.1. The blue bars denote the rankings for which there was no significant correlation, with a significance level $\alpha = 0.05$. The green bars denote the significant correlations.

reference thesaurus. The globally trained thesaurus performing better could simply mean that it was most similar to the reference thesaurus, without actually being the better thesaurus. This also applies to the general performance of the trained thesauri. Even though they showed a big discrepancy with the ground truth thesaurus, that does not have to mean that the trained thesauri perform poorly when used for query expansion.

This is especially relevant for the comparisons in this paper, since the ground truth thesaurus and the trained thesauri will naturally consist of different terms. The trained thesauri are namely purely based on the terms in the document space, whereas the manually constructed thesaurus is based on terms from concept lists and dictionaries, which might not actually appear as such in the collection. This was also reflected by the fact that the vocabularies of the trained thesauri did not contain all of the terms of the ground truth thesaurus.

Table 6 illustrates the problem of this discrepancy when using a reference thesaurus for evaluation. Although the automatically constructed thesauri make some clear mistakes, often in the form of linking to words with very similar usage (e.g. nationalities) or linking to words from very specific cases (e.g. linking *namaak* to *merk Colt*), they also contain related terms that are not found in the reference thesaurus (e.g. *groepsactie* and *collectieve actie*). In the latter case, the constructed thesauri are thus unfairly punished in the evaluation.

Given these limitations, a better approach would be to evaluate the thesauri directly on a query expansion task. Direct evaluation does not only remove the ambiguity of the performance evaluation, it also allows the thesauri to be compared to other query expansion

Table 5. An overview of the types of relations correctly retrieved by the three constructed thesauri, expressed in percentages of the total number of retrieved relations. Here $k = 1$ means that only the first element of the most similar terms is taken into account, $k = 5$ means only the first five elements, etcetera.

Retrieved Term Types	Global			Maximum			Average		
	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$
Narrow-term	7%	11%	12%	10%	11%	12%	9%	11%	12%
Broader-term	6%	10%	11%	9%	10%	10%	8%	9%	10%
Related-term	24%	26%	28%	25%	28%	30%	28%	29%	30%
Use	35%	28%	26%	28%	26%	24%	28%	25%	24%
Used-for	28%	25%	23%	29%	25%	24%	27%	25%	24%

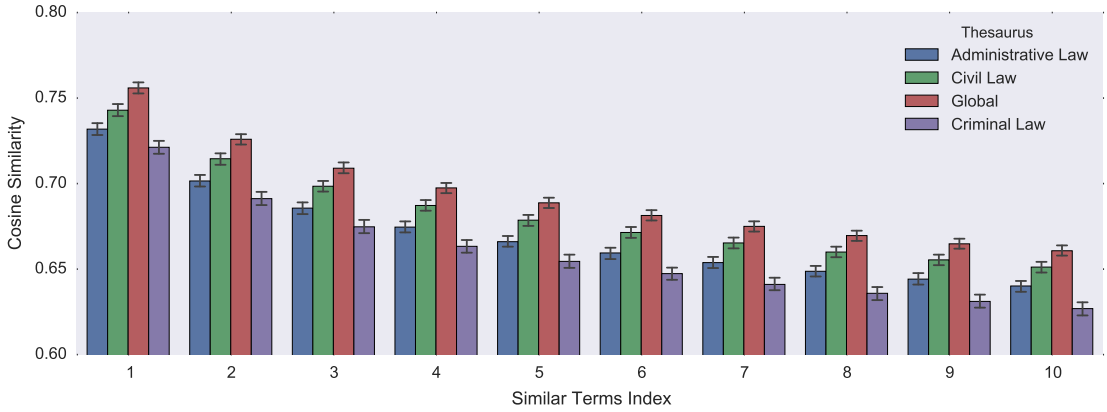


Figure 3. The mean cosine similarity scores for each rank of the most similar term rankings of the trained models. The error bars denote the standard deviation.

techniques. For a more complete overview of direct and indirect evaluation of thesauri see the paper written by Claveau and Kijak (2016a). Direct evaluation was unfortunately not possible for this experiment, because no relevance data or query logs were available.

5.2. Data Imbalance

After splitting the training data into multiple jurisdictions, the data was not equally balanced between the different jurisdictions (see Tables 2 and 3). A jurisdiction with less data might result in lower cosine similarity scores for that jurisdiction, since there is less text available to reinforce the context patterns of the words. This way the imbalance in the data could cause jurisdictions with less data to be unfairly underrepresented in the local thesauri.

The possible correlation between similarity scores and the size of the training data is partially supported by Figure 3. The figure shows that the similarity scores for criminal law are on average the lowest for every position in the rankings of most similar term, whereas the model based on all of the data consistently has the highest similarity scores. Surprisingly, the model trained on civil law actually has significantly higher scores than the other two local models, even though it only had the second most training data. Further research is required to confirm whether data imbalance effects the similarity scores and to explore methods to then compensate for the imbalance.

5.3. Further Research

For this experiment, the performance of the trained thesauri was evaluated for the one, five and ten most similar terms. Although the results showed the highest MAP@k scores for $k = 1$, more focused research has to be done to gain insight into the ideal number of similar terms that have to be used for the trained thesauri. This number will most likely differ between contexts in which the thesauri are used and as such be evaluated separately for specific contexts.

Since this paper strictly focused on comparing the performance of globally and locally trained thesauri using a ground truth thesaurus, it did not touch on the actual construction of thesauri from the models. For the

Table 6. Some examples of differences between the reference thesaurus and the automatically constructed thesauri. English translations of the Dutch terms are given between the parentheses

Term	Reference Thesaurus	Local Thesaurus	Global Thesaurus
Marokkanen (*Moroccans*)	allochtonen (*immigrants*)	Antillianen (*Antilleans*)	Turken (*Turks*)
Benelux	Comité van Ministers (*Committee of Ministers*)	BVIE (*Benelux Convention Intellectual Property*)	woord- beeldmerk (*word- figurative mark*)
XTC	ecstasy, MDMA	heroïne (*heroin*)	GHB
nabootsing (*imitation*)	namaak (*counterfeit*), imitatie (*imitation*)	merk Colt (*authentic Colt*)	replica (*replica*)
groepsactie (*group action*)	class action	collectieve actie (*collective action*)	collectieve actie (*collective action*)
anonieme melding (*anonymous report*)	Meld Misdaad Anoniem	anonieme tip (*anonymous tip*)	anonieme tip (*anonymous tip*)
opzegging (*notice*)	duurovereenkomst (*fixed-term agreement*)	beëindiging (*termination*)	ontbinding (*termination*)
natuurbeheer (*nature management*)	jacht (*hunt*), milieubeheer (*environmental management*)	subsidieregeling agrarisch (*agricultural subsidy*)	landschapbeheer (*landscape management*)
pesten (*bullying*)	school en criminaliteit (*school and crime*)	uitschelden (*calling names*)	stalken (*stalking*)
knowhow	industriële knowhow (*industrial knowhow*)	know how	know-how

comparison in this paper, it sufficed to select only the terms that were shared by the trained thesauri and the reference thesaurus. In practice though, selecting appropriate terms from the models is a crucial part of forming a thesaurus. A possible selection technique would be to use part-of-speech tagging to only select noun phrases. The models could also be compared to models trained on general text, as a way to only select terms that are specific to the legal domain.

Another challenging aspect of automatic thesaurus construction is automatically annotating the relations between terms. For this, more research is required to take the trained thesauri and identify these relations.

Finally, the techniques described in this paper could also be tested in different domains, to gain insight into whether the results carry over.

6. Conclusion

This study, first of all, set out to confirm that a thesaurus trained on the entire collection differs from a thesaurus trained on separate legal fields. The results showed a significant difference between the globally and locally trained Word2Vec embeddings in the automatic construction of thesauri. This difference can be attributed to the fact that relevant, but context specific uses of terms, might not be captured by the neural network, because they get overshadowed in the grand scheme. This generalization effect was also mentioned in previous papers by Diaz et al. (2016) and Roy et al. (2016).

As a follow up, this paper set out to answer whether local Word2Vec models created a better thesaurus than global models. This however proved not to be the case, unlike the initial expectation that the generalization effect would result in a poorer performance of the global thesaurus. The globally trained thesaurus actually outperformed both locally trained thesauri in the experiment.

Two methods to resolve conflicts in the concatenation of the local models were explored. The experiment showed that taking the maximum cosine similarity score consistently outperformed the average similarity score.

It is hard to draw definite conclusions from the experiments though, since all three thesauri performed poorly. The thesauri showed very low MAP@k and r-precision scores when retrieving relevant terms from

the ground truth thesaurus, which means that the trained thesauri retrieved a large number of irrelevant terms. In other words, there was a big discrepancy between the terms that were considered relevant by the reference thesaurus and the terms considered relevant by the trained thesauri.
This discrepancy was also reflected in the bias of the trained thesauri in favor of synonyms, when compared to the reference thesaurus. This bias stems from the assumption of Word2Vec, that related terms are used in similar contexts. Synonyms namely have a natural tendency to occur more often in similar contexts than broader, narrower or otherwise related terms.

References

Abdul-Jaleel, N., Allan, J., Croft, W. B., Diaz, F., Larkey, L., Li, X., Smucker, M. D., & Wade, C. (2004). Umass at trec 2004: Novelty and hard. *Online Proceedings of the 2004 Text Retrieval Conference*.

Bhogal, J., Macfarlane, A., & Smith, P. (2007). A review of ontology based query expansion. *Information processing & management*, *43*, 866–886.

Claveau, V., & Kijak, E. (2016a). Direct vs. indirect evaluation of distributional thesauri. *Proceedings of the International Conference on Computational Linguistics, COLING*.

Claveau, V., & Kijak, E. (2016b). Distributional thesauri for information retrieval and vice versa. *Language and Resource Conference, LREC*.

Diaz, F., Mitra, B., & Craswell, N. (2016). Query expansion with locally-trained word embeddings. *ACL '16*.

Hersh, W., Price, S., & Donohoe, L. (2000). Assessing thesaurus-based query expansion using the umls metathesaurus. *Proceedings of the AMIA Symposium* (pp. 344–348).

IJzereef, L., Kamps, J., & De Rijke, M. (2005). Biomedical retrieval: How can a thesaurus help? *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 1432–1448).

Lauser, B., Johannsen, G., Caracciolo, C., van Hage, W. R., Keizer, J., & Mayr, P. (2008). Comparing human and automatic thesaurus mapping approaches in the agricultural domain. *Metadata for Semantic and Social Applications: Proceedings of the International Conference on Dublin Core and Metadata Applications* (pp. 43–53).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* (pp. 3111–3119).

Minker, J., Wilson, G. A., & Zimmerman, B. H. (1972). An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Information Storage and Retrieval*, *8*, 329–348.

Navigli, R., & Ponzetto, S. P. (2012). Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, *193*, 217–250.

Roy, D., Paul, D., Mitra, M., & Garain, U. (2016). Using word embeddings for automatic query expansion. *Neu-IR '16 SIGIR Workshop on Neural Information Retrieval*.

Tudhope, D., Binding, C., Blocks, D., & Cunliffe, D. (2006). Query expansion via conceptual distance in thesaurus indexed collections. *Journal of Documentation*, *62*, 509–533.

van Netburg, C. J., & van der Weijde, S. Y. (2015). Justitiethesaurus 2015.

Identifying writing tasks using sequences of keystrokes

Rianne Conijn M.A.CONIJN@TILBURGUNIVERSITY.EDU
Menno van Zaanen MVZAANEN@TILBURGUNIVERSITY.EDU
Department of Communication and Information Sciences, Tilburg University, The Netherlands

Keywords: keystroke analysis, classification, writing processes, writer identification

Abstract

The sequences of keystrokes that are generated when writing texts contain information about the writer as well as the writing task and cognitive aspects of the writing process. Much research has been conducted in the area of writer identification. However, research on the analysis of writing processes based on sequences of keystrokes has received only a limited amount of attention. Therefore, in this study we try to identify properties of keystrokes that indicate cognitive load of the writing process. Moreover, we investigate the influence of these properties on the classification of texts written during two different writing tasks: copying a text and free-form generation of text. We show that we can identify properties that allow for the correct classification of writing tasks, which at the same time do not describe writer-specific characteristics. However, some properties are the result of an interaction between the typing characteristics of the writer and the writing task.

1. Introduction

Students' activities in online learning systems can provide useful information about their learning behavior. Educational data mining focuses on the use of data from learners and their context to better understand how students learn, to improve educational outcomes, and to gain insight into and explain educational phenomena (Romero & Ventura, 2013). Data can be collected from different sources, such as online learning systems, student administration, and questionnaires. This results in data from multiple contexts, over different time periods, ranging from low to high granularity. In this study, we analyze fine-grained data: keystroke data from a writing task.

In the literature, two different goals can be distinguished in the analyses of keystroke data: authentication or identification of writers, and determination of writing processes. Keystrokes have mainly been used for the former (Longi et al., 2015; Karnan et al., 2011). In the field of educational data mining, the authentication and identification of writers is used, for example, for authentication in online exams (Gunetti & Picardi, 2005) or for the identification of programmers (Longi et al., 2015). These studies mainly focus on the typing or motor processes, since these are considered unique per person. The majority of studies focus on statistical properties, such as mean, standard deviation, and Euclidean distance of three attributes: keystroke duration, keystroke latency, and digraph duration (Karnan et al., 2011). These features can be used to identify and authenticate writers to a large extent, with accuracies up to 99% (Tappert et al., 2009). These high accuracies show that keystroke logs contain much information that denotes writer-specific characteristics.

Yet, keystroke data also includes other information, denoting the writing process itself. The determination of these writing processes has received less attention. This might be due to the fact that keystrokes are not clear measures of the underlying writing processes (Baaijen et al., 2012). The data need to be preprocessed and analyzed in a way such that it provides meaningful information to be used by students and teachers for improving learning and teaching. Therefore, this study explores the writing processes derived from students' keystrokes.

Some studies already investigated the possibilities of determining writing processes using keystrokes. Baaijen et al. (2012) analyzed keystroke data from 80 participants during a 30-minute writing task. The rela-

tion between pauses, bursts, and revisions were analyzed. Using these features, text production could be distinguished from revisions. Revision bursts were shorter than new text production bursts. In another writing task, keystroke data from 44 students during a 10-minute essay was collected to determine emotional states (Bixler & D'Mello, 2013). Four feature sets were used: total time, keystroke verbosity (number of keys and backspaces), keystroke latency, and number of pauses (categorized by length). All feature sets combined could classify boredom versus engagement with an accuracy of 87%. Keystroke data have also been analyzed in programming tasks, to determine performance. Thomas et al. (2005) analyzed keystroke data from 38 experienced programmers and 141 novices in a programming task. Keystroke latencies and key types were found related to performance. Key latencies (within and before or after a word) were found negatively correlated with performance. Additionally, it was found that experienced programmers used more browsing keys and were faster in pressing those.

These studies show that keystrokes do not only differ due to writer-specific characteristics (which is used in authentication and identification), but also because of differences in revisions and text production, emotional states, and level of experience. Whereas the differences in writer-specific properties may be due to physical differences and differences in typing style, the differences in writing properties are expected to come from differences in cognitive processes required. Indeed, keystroke duration and keystroke latencies are often seen as an indicator of cognitive load (Leijten & Van Waes, 2013). As different tasks lead to differences in cognitive load, we may find these differences using different writing tasks. However, existing studies do not compare differences in keystrokes between tasks. Therefore, in the current study, the writing processes in two different tasks are compared: writing a free-form text versus a fixed text (copying a text). Here we assume that writing a free-from text requires a different cognitive load than writing a fixed text, resulting in differences in the keystroke data.

Having knowledge of the cognitive load while producing a text may provide useful information, for example, for teachers. Currently, teachers often only have access to the final writing product for evaluation purposes. This does not provide insight in what students did during the writing process. Insight in students' writing behavior or cognitive load during an assignment may trigger the teacher to further investigate this behavior and adapt the task or provide personalized feedback on the writing process.

To identify properties of keystrokes that indicate the cognitive load of the writing process, an open dataset is used, which has been used for writer identification. In a previous study, it was already shown that keystroke data differed between free-form and fixed text (Tappert et al., 2009). However, these differences were not made explicit nor evaluated. Therefore, in the current study, we analyze which features differ within the keystrokes of free-form versus fixed text using three different feature sets. As an evaluation, the differences found between fixed and free-form text are used to classify text as being either fixed or free-form text. This is done using all possible combinations of the different feature groups, to determine which feature group is most useful for the classification. At the same time, since we are not interested in the writer-specific information, the properties should not allow for an accurate identification of the actual writer.

2. Method

2.1. Data

Data used in the current experiments has been taken from the Villani keystroke dataset (Tappert et al., 2009; Monaco et al., 2012). The Villani keystroke dataset consists of keystroke data collected from 142 participants in an experimental setting. Participants were free to chose to copy a fable, a fixed text of 652 characters, or to type a free-form text, an email of at least 650 characters. Participants could copy the fable multiple times and could also type multiple free-form texts. Since typing the texts was not mandatory, not all participants typed both a free-form text and a fixed text. In total, 36 participants typed both at least one fixed text and one free-form text, resulting in keystroke data of 338 fixed texts and 416 free-form texts. The other 106 participants only wrote either free-form or fixed texts, resulting in a further 21 fixed texts and 808 free-form texts. The keystroke data consisted of timestamps for each key press and key release and the corresponding key code. More information about the dataset and the collection of the dataset can be found in Tappert et al. (2009). In this research, we only use the data of participants who created both text types.

2.2. Data processing

First, for all keystrokes, the type of key was derived: letter, number, browse key (e.g., LEFT, HOME), punctuation key, correction key (BACKSPACE, DELETE), and other (e.g., F12). The time between a key release and the subsequent key press (keystroke latency or key pause time) was calculated. Thereafter, the type of pause between the key was derived using

the key types. Six pause types were identified: pause before word (after SPACE, before letter or number), within word (between letters or numbers), after word (after letter or number, before SPACE), before correction, within correction, and after correction.

Lastly, words were identified as the letters and numbers between two SPACE keys. For all words the word length (number of letters and numbers) and the word time was calculated (time from key press of the first character until time of the key release of the last character). For further analyses on the word length and word time, only words without corrections were included. The use of corrections within a word would have a significant influence on the time of typing. Additionally, since a BACKSPACE or DELETE key can be used to remove multiple characters, it is hard to determine word length if corrections are made within the word.

Figure 1 shows the measurement of the timing of the different types of pauses. Given that the writer types the words "the book" with two incorrect letters after the SPACE key ("do"), which are corrected using two BACKSPACEs, the key presses and releases per key are illustrated in the second row. The following rows each depict which periods between key releases and key presses are measured for that type of pause. For instance, the pause before the word "the" and the pause between the SPACE key and the letter "d" are counted as pause before word type (third row). The last row indicates the timing used to compute the word length. In this case, the word "the" is identified (which has a length of three characters). The word "book" is not used, as it contains corrections.

After data enrichment, the three groups of features were computed: pause times, corrections, and word length. For all six different types of pause times (see Figure 1), the normalized average pause times were calculated by dividing the average pause time of each type by the overall average pause time over all types. Additionally, the normalized standard deviations of the pause times were calculated. In total, this resulted in 12 different features related to pause time. For the corrections, two features were calculated: the total number of corrections and the percentage of words with corrections. Lastly, four features related to word length were computed: the average time and standard deviation for short words (having less than four characters) and the average time and standard deviation for long words (consisting of between 9 and 13 characters). All four features were normalized using the average time and standard deviation of all words.

Obviously, the keystroke sequences contain much more

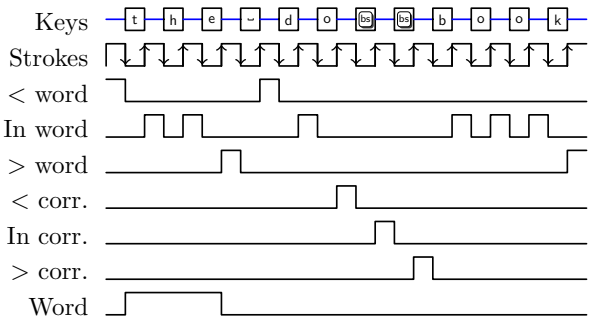


Figure 1. Measurement of timing of pauses of "the book" with two corrections using the backspace key (bs). "<" means before, ">" means after, "corr." stands for correction. The last row indicates the time of the word "the" (of length three characters). The word "book" is not used, as it contains corrections.

information than what we extracted here. The selection of these features was made in order to reveal as little as possible about the actual text being typed. Actual key code information, for instance, is not used as that information should be quite consistent between the fixed texts.

For the statistical analyses and training of the models, only data from the 36 participants who typed both fixed and free-form texts were included. From these, four texts were excluded, because they consisted of less than five words and inspection showed that these texts were random key strokes. Thereafter, 750 texts (338 fixed and 412 free-form) remained for analyses.

2.3. Analyses

To identify the relationship between keystroke information and cognitive load in writing tasks, two types of analyses were used: statistical analyses and model evaluation. First, paired t-tests were conducted to determine whether differences were found between the features in the fixed and free-form texts of participants.

Thereafter, support vector machines were trained to classify texts as being fixed or free-form. Support vector machines were trained for all combinations of the three feature groups (pause times, corrections, and word length), resulting in a total of seven models. The radial base function was used as kernel ('svmRadial' from the 'caret' package in R (Kuhn, 2016)). The data was trained using 10-fold cross-validation. Grid search was used during training (with a tuning part held aside from the testing) to optimize the parameters σ and cost. The average accuracy were calculated as performance measures. Since the groups were not equally distributed, the average κ was also calculated. The

Table 1. Descriptive statistics and paired t-tests of features in fixed and free-form text (N = 36). *=$p < .05$, **=$p < .01$, ***=$p < .001$.

Feature	Fixed text		Free-form text	
	M	*S.D.*	*M*	*S.D.*
Total time ***	376	100	432	129
# Keys **	703	54.6	749	57.5
# Corrections ***	21.7	13.6	35.3	16.4
% Words corrected ***	0.08	0.05	0.13	0.06
Average pause time before word **	1.17	0.18	1.25	0.18
S.D. pause time time before word *	1.04	0.25	1.16	0.28
Average pause time within word ***	0.86	0.09	0.78	0.07
S.D. pause time within word ***	0.60	0.21	0.40	0.15
Average pause time after word **	0.91	0.15	0.84	0.17
S.D. pause time after word **	0.79	0.29	0.61	0.30
Average pause time before correction	2.49	0.76	2.40	1.52
S.D. pause time before correction *	1.50	0.55	1.22	0.51
Average pause time within correction	0.95	0.61	0.82	0.16
S.D. pause time within correction	0.48	0.30	0.44	0.24
Average pause time after correction	1.75	1.60	2.16	3.29
S.D. pause time after correction	1.14	0.29	1.03	0.30
Average short word time **	0.52	0.05	0.48	0.09
S.D. short word time	0.33	0.09	0.32	0.11
Average long word time	2.87	0.26	2.98	0.71
S.D. long word time	1.14	0.23	1.16	0.28

κ corrects for random guessing, by comparing the observed accuracy with the expected accuracy (chance):

$$\kappa = \frac{observed\ accuracy - expected\ accuracy}{1 - expected\ accuracy}$$

Additionally, a one-way ANOVA with Tukey post-hoc test was used to determine whether the models differed significantly in accuracy.

Lastly, since we focus on the writing process, the learned models should preferably not be able to classify personal writer-specific characteristics. Thus, the learned model should perform really badly when classifying writers. Therefore, as an additional evaluation support vector machines were trained to classify the writers. The best σ and cost values from the models classifying fixed versus free-form text were used. Again, the average accuracy and κs were calculated using the same folds in 10-fold cross-validation.

3. Features measuring cognitive load

Paired t-tests were used to determine which features differed significantly between fixed and free-form text created by the same writer. This is assumed to provide insight in which features are indicative of cognitive load. The results can be found in Table 1. Note that we use both the mean as well as the standard deviation (S.D.) within a text as features (and both types of features have their own standard deviations per document). In the table, the descriptive statistics of these features can also be found.

It was found that fixed texts consisted of significantly fewer keystrokes compared to free-form texts (703 versus 749). Although the fixed text consisted of 652 characters, the mean number of keystrokes was 703. This can partly be explained by the fact that sometimes multiple keys are needed to type one character (e.g., SHIFT + character to type a capital letter). Additionally, this can indicate that the participants made typos and fixed those, requiring BACKSPACE or DELETE keystrokes. Indeed, it was shown that corrections were made in 740 of the 750 sessions. The free-form texts contained more corrections and a higher percentage of words with at least one correction, compared to the fixed texts. Lastly, the participants were faster in typing the fixed text compared to the free-form text. All these findings provide some evidence that typing the free-form text requires a different cognitive load.

Several features were analyzed to determine where significant differences in pause duration between the text types were found. Specifically, we investigated the differences between the pauses before, after, and within words and corrections. Since the free-form and fixed texts differed in total length and time, timing of key pauses were normalized based on the average time per key pause in that session. It was found that writers

spend more of their writing time on pauses before a word and less time on pauses within a word or after a word in free-form text, compared to fixed texts. Additionally, the standard deviation of pauses within and after a word was significantly lower for free-form texts compared to fixed texts. For the pauses before words, the opposite was found: when typing free-form text, a larger proportion of pause time was spent before a word, compared to fixed texts. Moreover, the standard deviations were larger. This may be because the writer will need to think (longer) about which word to type in free-form text, which is not needed for fixed texts.

For the key pause times before, after, and within corrections, no significant differences were found between free-form and fixed texts. The only exception is the standard deviation of key pause time before corrections: free-form texts lead to a larger standard deviation for key pause time before corrections compared to fixed texts.

When comparing the average word time between the two types of text, participants were faster in typing short words (consisting of less than four letters) in free-form text compared to fixed text. This indicates that in free-form text, of all words, less time is devoted to short words, compared to fixed text. No significant differences were found between fixed and free-form texts for the average word time for long words (8–13 letters). Additionally, no significant differences were found in the standard deviation of time on short and long words between the text types.

4. Model evaluation

4.1. Classifying fixed versus free-form text

To measure the effect of the different groups of features, we trained support vector machines and measured how well they could distinguish between a fixed and a free-form text. The models were trained using all combinations of three different feature groups: average and standard deviations of the pause times (Pauses); correction information (Correction); and average and standard deviation of short and long word typing time (Words). The accuracies and κs of all seven models for classifying fixed and free-form text can be found in Table 2.

The results show that all feature groups are useful for the classification of fixed versus free-form text. The feature group of the key pause times led to best accuracy (73.9% with a κ of 0.465, or approximately 47% above chance), compared to the other individual feature groups. Not surprisingly, the combination of all feature groups yielded the overall highest accuracy: 78.1% with a κ of 0.551. A one-way ANOVA showed that the seven models differed significantly in accuracy ($F(6, 63) = 4.728, p < .001$). Using two feature groups was always better than using only correction features or word length features. However, the combination of all feature groups did not lead to a significantly higher accuracy than the pause time features alone. Thus, the word length and the corrections features did not have much additional value next to the pause time features for classifying fixed versus free-form text.

Table 2. Accuracies and κs of support vector machine models on the different feature groups that classify fixed versus free-form text.

feature group	Fixed vs. free Accuracy	κ
Pauses	0.739	0.465
Correction	0.689	0.368
Words	0.687	0.370
Pauses, Correction	0.763	0.513
Correction, Words	0.767	0.522
Pauses, Words	0.739	0.465
Pauses, Correction, Words	0.781	0.551

Table 3. Accuracies and κs of support vector machine models on the different feature groups that classify writers (36 classes).

Feature group	Writer Accuracy	κ
Pauses	0.248	0.223
Correction	0.091	0.065
Words	0.073	0.046
Pauses, Correction	0.311	0.287
Correction, Words	0.121	0.095
Pauses, Words	0.239	0.215
Pauses, Correction, Words	0.291	0.267

4.2. Classifying writers

To determine whether the learned models did not include any writer-specific characteristics, models with the same settings as the models that classify text types were trained and tested to classify writers. The results of these experiments can be found in Table 3. Similarly to the models classifying fixed versus free-format text, the key pause time features led to a higher accuracy (24.8% and $\kappa = 0.223$) than the correction and word length features. The correction and word length features resulted in the lowest accuracies: 9.1% and

7.3%, respectively. The model with both correction and pause time features led to the highest accuracy: 31.1% with a κ of 0.267. Although this is a reasonably low accuracy, the model clearly outperforms the model based on chance for the 36-class classification ($1/36 = 2.8\%$). This means that some writer-specific characteristics are encoded within the feature groups that are used in these models, which in this case is unwanted.

A one-way ANOVA showed that the seven models differed significantly in accuracy ($F(6, 63) = 37.43, p < .001$). Interestingly, the accuracies for the corrections and words feature groups combined are significantly lower than the accuracy of all models that include pause time features. This indicates that the corrections and word length features include fewer writer-specific characteristics than the pause time features.

5. Discussion

Keystroke data include both writer-specific information and information about the writing processes. In this study, we focused on the writing processes and aimed to identify properties of keystrokes that indicate the cognitive load of the writing process. In order to do this, keystrokes of two different writing tasks were analyzed, which are assumed to differ in cognitive load: copying a text (fixed text) and writing a free-form text.

Our first analysis showed that several features extracted from the keystroke data differed significantly between the fixed and free-form texts of a writer. These findings support previous work which showed that keystrokes differ for different (types of) text entered (Gunetti & Picardi, 2005; Tappert et al., 2009). As an extension, we also identified which features differed and how these differed. When typing free-form text, the pauses before a word were longer, while the pauses within or after a word were shorter, compared to typing fixed text. This might indicate that the participants were thinking about the next word to type in the free-form text before they typed the word, while writers in the fixed text situation could immediately copy it as it was provided for them. Thus, differences in cognitive load may be identified in the pauses before words.

As an evaluation, we showed that the differences in keystroke information can be used to classify fixed and free-form text. Using a support vector machine, the key pause time features (which measure time spent between key releases and key presses) were found to lead to the highest accuracies for the text identification task. Adding the corrections and word length features did not lead to significantly higher accuracies, showing that the word length and corrections features do not add much information in addition to the pause times. When all feature groups were included, 78.1% accuracy was reached. Although this accuracy is reasonably high, it also shows there is still some room for improvement. Especially considering that classifying writers, being a more complex classification problem with more classes, have shown accuracies up to 99% (Tappert et al., 2009).

Since we aimed to identify features related to the writing process, we wanted to exclude writer-specific information. In other words, the models should perform badly when classifying the writer. To test this, we tried to classify the writers with the same settings as used in the writing task classification for the support vector machine models. The lowest accuracy, while using information from the keystroke sequences were found when using word length features only (7.3%). This corresponds to an accuracy of 68.7% on the text type classification task. The highest accuracy (31.1%) was obtained with both pause time and correction features (corresponding to 76.3% accuracy on the text type classification task, which is close to the highest accuracy on that task: 78.1%). Even though the accuracies on writer classification are higher than chance, it is much lower than the 90%–99% accuracy reached in other studies (Longi et al., 2015; Tappert et al., 2009). Thus, the feature groups that have been extracted, actually contain mostly information related to the writing task and not to the writer-specific characteristics.

Interestingly, especially the corrections and word length features showed low accuracies on classifying writers. Thus, these feature groups contained little information about individual typing characteristics. Adding additional information to improve the quality of the text type classification task, also increases accuracy of the writer classification task. For example, if we add the key pause features, the accuracy of the text type task increases, but the writer identification accuracies also increase. In other words, key pause properties contain useful information for the text type classification task, but also contain information that allows for the identification of the writer, which is unwanted in this case.

There are at least three directions for future work. Firstly, future work could try to improve the accuracy on task classification, while not improving the accuracy on writer identification. Additional features or feature groups could be identified, such as bursts

ending in a pause (P-bursts) or ending in a revision (R-bursts) (Baaijen et al., 2012). In addition, the key pause feature group seems to contain useful information, but these features should be modified in order to remove any writer-specific properties. Alternatively, other machine learning algorithms, such as neural networks, may be tried to achieve higher accuracies.

Secondly, a wider range of writing tasks could be considered. For example, semi-fixed tasks with specific task descriptions (e.g., writing a sorting algorithm) can be investigated to determine whether differences between tasks that are more similar can also be distinguished. In this way, we may identify which tasks require more cognitive load and in which properties of the process of typing this effort can be found. This information can be used to improve the writing task instruction or to provide feedback on the writing process to the learner during the writing task (see also Poncin et al., 2011; Kiesmueller et al., 2010).

Lastly, this study assumed that the differences in keystrokes provide an indication of cognitive load. However, we did not actually measure the cognitive load. Future work could explicitly measure the cognitive load during the task, for example by using a secondary task, or a questionnaire (Paas et al., 2003). In this way, the problem could be approached as a regression problem rather than a classification task.

6. Conclusion

To conclude, this research has shown that keystroke data can be used to identify differences in writing tasks, which we believe require different cognitive load. Additionally, we showed which feature groups (key pause, correction, and word length) have an influence on the performance. In particular, the word length and correction feature groups led to a good performance on the writing task classification task and a low performance on the writer identification task. The key pause feature group increases the performance on the text classification task, but the performance on the writer classification task also increases (which is unwanted, as the key pause features also include writer-specific properties).

Having insight in these features which identify writing processes and cognitive load can be useful for improving learning and teaching. For example, in this way, teachers can also get insight in the writing process, instead of only the product of writing. This can be useful for adapting the course materials or providing (personalized) feedback.

References

Baaijen, V. M., Galbraith, D., & de Glopper, K. (2012). Keystroke analysis: Reflections on procedures and measures. *Written Communication*, *29*, 246–277.

Bixler, R., & D'Mello, S. (2013). Detecting boredom and engagement during writing with keystroke analysis, task appraisals, and stable traits. *Proceedings of the 2013 international conference on Intelligent user interfaces* (pp. 225–234).

Gunetti, D., & Picardi, C. (2005). Keystroke analysis of free text. *ACM Transactions on Information and System Security (TISSEC)*, *8*, 312–347.

Karnan, M., Akila, M., & Krishnaraj, N. (2011). Biometric personal authentication using keystroke dynamics: A review. *Applied Soft Computing*, *11*, 1565–1573.

Kiesmueller, U., Sossalla, S., Brinda, T., & Riedhammer, K. (2010). Online identification of learner problem solving strategies using pattern recognition methods. *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 274–278).

Kuhn, M. (2016). *caret: Classification and regression training*. R package version 6.0-73.

Leijten, M., & Van Waes, L. (2013). Keystroke logging in writing research: Using inputlog to analyze and visualize writing processes. *Written Communication*, *30*, 358–392.

Longi, K., Leinonen, J., Nygren, H., Salmi, J., Klami, A., & Vihavainen, A. (2015). Identification of programmers from typing patterns. *Proceedings of the 15th Koli Calling Conference on Computing Education Research* (pp. 60–67).

Monaco, J. V., Bakelman, N., Cha, S.-H., & Tappert, C. C. (2012). Developing a keystroke biometric system for continual authentication of computer users. *Intelligence and Security Informatics Conference (EISIC), 2012 European* (pp. 210–216).

Paas, F., Tuovinen, J. E., Tabbers, H., & Van Gerven, P. W. (2003). Cognitive load measurement as a means to advance cognitive load theory. *Educational psychologist*, *38*, 63–71.

Poncin, W., Serebrenik, A., & van den Brand, M. (2011). Mining student capstone projects with frasr and prom. *Proceedings of the ACM international*

conference companion on Object oriented programming systems languages and applications companion (pp. 87–96).

Romero, C., & Ventura, S. (2013). Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *3*, 12–27.

Tappert, C. C., Villani, M., & Cha, S.-H. (2009). Keystroke biometric identification and authentication on long-text input. *Behavioral biometrics for human identification: Intelligent applications*, 342–367.

Thomas, R. C., Karahasanovic, A., & Kennedy, G. E. (2005). An investigation into keystroke latency metrics as an indicator of programming performance. *Proceedings of the 7th Australasian conference on Computing education-Volume 42* (pp. 127–134).

Increasing the Margin in Support Vector Machines through Hyperplane Folding

Lars Lundberg LARS.LUNDBERG@BTH.SE
Håkan Lennerstad HAKAN.LENNERSTAD@BTH.SE
Eva Garcia-Martin EVA.GARCIA.MARTIN@BTH.SE
Niklas Lavesson NIKLAS.LAVESSON@BTH.SE
Veselka Boeva VESELKA.BOEVA@BTH.SE
Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden

Keywords: support vector machines, geometric margin, hyperplane folding, hyperplane hinging, piecewise linear classification.

Abstract

We present a method, called hyperplane folding, that increases the margin in a linearly separable binary dataset by replacing the SVM hyperplane with a set of hinging hyperplanes. Based on the location of the support vectors, the method splits the dataset into two parts, rotates one part of the dataset and then merges the two parts again. This procedure increases the margin in each iteration as long as the margin is smaller than half of the shortest distance between any pair of data points from the two different classes. We provide an algorithm for the general case with n-dimensional data points. A small experiment with three folding iterations on 2-dimensional data points shows that the margin does indeed increase and that the accuracy improves with a larger margin, i.e., the number of misclassified data points decreases when we use hyperplane folding. The method can use any standard SVM implementation plus some additional basic manipulation of the data points, i.e., splitting, rotating and merging.

1. Introduction

Support Vector Machines (SVMs) find the separating hyperplane that maximizes the margin to the data points closest to the hyperplane. The main idea with SVM is that, compared to a small margin, a large margin reduces the risk of misclassification of unknown points. In this paper, we present a method that in many cases increases the SVM margin in a linearly separable dataset.

Consider a set of points $S \subset \mathbb{R}^n$ that can be separated linearly in two subsets S_+ and S_-. By choosing a hyperplane P that maximizes $min_{v \in S}\ dist(v, P)$, i.e., the minimal distance to all points in S, we obtain a separating hyperplane that has the same distance to all points in a certain set $V_- \cup V_+$, where $V_- \subset S_-$ and $V_+ \subset S_+$. The set $V_- \cup V_+$ is the set of support vectors, where both V_- and V_+ are non-empty. The margin of this hyperplane P is $m(P) = dist(v, P)$ for any support vector $v \in V_- \cup V_+$. The largest possible margin for any separating surface, linear or not, is trivially $M(S) = min_{v \in S_+, u \in S_-} |v - u|/2$. For any surface we define the margin as $m(P) = min_{v \in S}\ dist(v, P)/2$. A surface that fulfills $m(P) = M(S)$ is called a *maximal margin surface*.

If the set of support vectors $V_- \cup V_+$ contains two elements only, we have $m(P) = M(S)$, and the starting hyperplane is a maximal margin surface. This case can happen also for more than two support vectors, but it is not the generic case.

In the generic case we have $3 \leq |V_- \cup V_+| \leq n+1$, for which we construct a separating folded surface, composited by different hyperplanes, where the margin is normally larger than $m(P)$.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 introduces a hyperplane folding algorithm for the 2-dimensional case. Section 4 discusses higher dimensions and proposes the algorithm for the general case. Section 5 presents the initial evaluation of the proposed algorithm for the 2-dimensional case and further discussion. Section 6 is devoted to conclusions and future work.

Preliminary work. Under review for Benelearn 2017. Do not distribute.

2. Related Work

Support vector machines (SVMs) originate from research conducted in statistical learning theory (Vapnik, 1995). SVMs are used in supervised learning, where a training set consisting of n-dimensional data points with known class labels is used for predicting classes in a test set consisting of n-dimensional data points with unknown classes.

Rigorous statistical bounds are given for the generalisation of hard margin SVMs (Bartlett, 1998; Shawe-Taylor et al., 1998). Moreover, statistical bounds are also given for the generalisation of soft margin SVMs and for regression (Shawe-Taylor & Cristianini, 2000).

There has been work on different types of piecewise linear classifier based on the SVM concept. These methods split the separating hyperplane into a set of hinging hyperplanes (Wang & Sun, 2005). In (Yujian et al., 2011), the authors define an algorithm that uses hinging hyperplanes to separate nonintersecting classes with a multiconlitron, which is a combination of a number of conlitrons, where each conlitron separates "convexly separable" classes. The multiconlitron method cannot benefit directly from experience and implementations of SVMs. Conlitrons and multiconlitrons need to be constructed with new and complex techniques (Li et al., 2014). The hyperplane folding approach presented here is a direct extension of the standard (hard margin) SVM (soft margin extensions are relatively direct and discussed later). As a consequence, hyperplane folding can benefit from existing SVM experience and implementations.

A piecewise linear SVM classifier is presented in (Huang et al., 2013). That method splits the feature space into a number of polyhedra and calculates one hinging hyperplane for each such polyhedron. Some divisions of the feature space will increase the margin in hard margin SVMs. However, unless one has detailed domain knowledge, there is no way to determine which polyhedra to select to improve the margin. The authors recommend basic approaches like equidistantly dividing the input space into hypercubes or using random sizes and shapes for the polyhedra. Based on the support vectors, the hyperplane folding method splits the feature space into two parts (i.e., into two polyhedra) in each iteration. Without any domain knowledge, the method guarantees that the split results in an increase of the margin (except for very special cases). As discussed above, hyperplane folding can directly benefit from existing SVM experience and implementations, which is not the case for the method presented by Huang et al.

As stated in (Cortes & Vapnik, 1995), SVMs combine three ideas: the idea of optimal separating hyperplanes, i.e., hyperplanes with the maximal margin, the idea of so called kernel tricks that extends the solution space to include cases that are not linearly separable, and the notion of so called soft margins to allow for errors in the training set.

If we assume that a data point in the test set can be at most a distance x from a data point belonging to the same class in the training set, then it is clear that we can only guarantee a correct classification as long as x is smaller than the margin. As a consequence, the optimality of the separating hyperplane guarantees that we will correctly classify any data point in the test set for a maximum x. This is very similar to error correcting codes that maximize the distance between any pair of code words (Lin & Costello, 1983). The hyperplane folding approach presented here increases the margin thus guaranteeing correct classification of test set data points for larger x.

SVMs are also connected to data compression codes. In (von Luxburg et al., 2004), the authors suggest five data compression codes that use an SVM hyperplane approach when transferring information from a sender to a receiver. The authors show that a larger margin in the SVM leads to higher data compression, and that the data compression can be improved by exploring the geometric properties of the training set, e.g., if the data points in the training set are shaped as an ellipsoid rather than a sphere. The hyperplane folding approach also uses geometric properties of the training set to improve the margin.

The kernel trick maps a data point in n dimensions to a data point in a (much) higher dimension, thus increasing the possibility to linearly separate the data points (Hofmann et al., 2008). The hyperplane folding approach presented here does some remapping of data points but it does not change the dimension of the data points.

3. Hyperplane Folding

In this section, we introduce the hyperplane folding algorithm for the 2-dimensional case. Higher dimensions will be discussed in Section 4.

Let us consider a standard SVM for fully separable binary classification set S with a separating hyperplane (the thick blue line in Figure 1) and a margin d. If we assume that each data point is represented with (very) high resolution, the probability of having more than three support vectors is arbitrarily close to zero in the 2-dimensional case. Therefore, in the current context we only need to consider the cases with two or three support vectors.

As it was mentioned in the introduction, we consider only the case $|V_- \cup V_+| = 3$ in the 2-dimensional scenario, because we already have a maximal margin if $|V_- \cup V_+| = 2$. Without loss of generality we assume that $|V_+| = 2$ and $|V_-| = 1$, and we refer to the point in V_- as the primary

support vector.

As a first step in our method, we split the dataset into two parts by identifying a splitting hyperplane, which in two dimensions is the line that is normal to the hyperplane and that passes through the prime support vector (see Figure 2). When splitting into two datasets, the prime support vector is included in both parts of the dataset.

The two parts of the dataset define one SVM each (see Figure 3), producing one separating hyperplane for each part of the dataset, where both margins are normally larger than the initial margin. We assume that the two new hyperplanes intersect with an angle α.

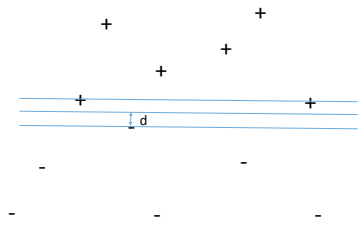


Figure 1. A separate dataset with three support vectors.

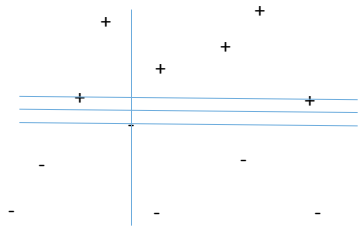


Figure 2. Splitting the dataset into two parts.

The folding takes place in the second step of our method, where all points in one of the two data sets are rotated the angle α around the intersection point, aligning the two new separating hyperplanes. We rotate the part with the largest margin. In Figure 4 we show the case when we rotate the data points in the right part.

After rotating, the data points in the two parts are joined into a new dataset of the same size as the original dataset. The new margin is the smallest of the margins of the two

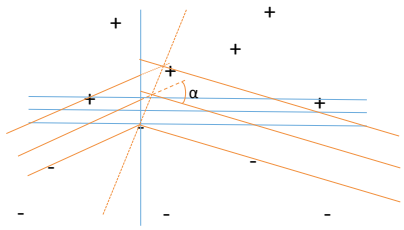


Figure 3. The two SVMs after splitting the dataset.

separate sets, which is larger than the initial margin, and has in general other support vectors than before (see Figure 5). If there are three support vectors in the new dataset, the same procedure can be repeated.

A detail explanation of the proposed hyperplane folding algorithm for the 2-dimensional case is given below.

Algorithm: Hyperplane Folding for the 2-dimensional Case

1: Run the standard SVM algorithm on the dataset $S \subset \mathbb{R}^2$.

   The output of SVM algorithm is:

   - The equation of a separating hyperplane
   - The support vectors $V_- \cup V_+$
   - The margin d

2: **if** $|V_- \cup V_+| = 2$ **then** terminate [1]

3: **if** the determined number of hyperplane folding iterations is reached **then** terminate

4: Select the primary support vector (one from the class with only one support vector).

5: Split S along a line (a splitting hyperplane) that is orthogonal to the separating hyperplane and that passes through the primary support vector.

6: Duplicate the primary support vector to both subsets of the data points.

7: Run the standard SVM algorithm separately on the two subsets, thus obtaining two separating hyperplanes.

[1] In the current context we assume that there can only be two or three support vectors. The case with more support vectors is discussed in Section 4.

8: Calculate the angle α between the two separating hyperplanes, and the intersection point between them, i.e. the folding point.

9: Remove the primary support vector from the subset with the largest margin.

10: Rotate the remaining data points in that subset an angle α around the folding point.

11: Merge the two subsets. The new splitting hyperplane has a larger margin than d.

12: **goto** step 1

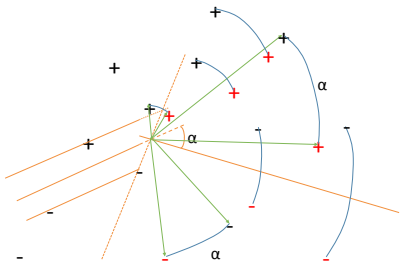


Figure 4. Rotating the data points in the right part of the dataset. Red data points in the original dataset are rotated with an angle α, counter clockwise, to new locations.

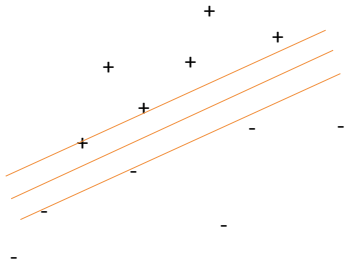


Figure 5. The dataset after rotation. The new dataset has $|V_+| = 1$ and $|V_-| = 2$ and a larger margin than the original dataset.

4. Higher Dimensions

In this section, we discuss higher dimensions. If we assume that each point is represented with (very) high resolution, there will in general be at most n+1 support vectors in an n dimensional space, $|V_-| \geq 1$ and $|V_+| \geq 1$.

We start by considering the case with three dimensions, i.e., $n = 3$. Again, we cannot do anything if we only have two support vectors, because then the starting hyperplane has maximal margin. In the case of three or four support vectors we can, however, increase the margin except in special cases. In order to simplify the discussion we assume that the separating hyperplane is parallel to the xz-plane, which can be achieved by rotating the data set.

In the case $|V_- \cup V_+| = 4$ there are three different cases: $\{|V_+| = 3, |V_-| = 1\}$, or $\{|V_+| = 2, |V_-| = 2\}$, or $\{|V_+| = 1, |V_-| = 3\}$. In either case we consider a line that passes through one pair of support vectors from the same class. This line is parallel to the separating hyperplane, since all support vectors have the same distance to this hyperplane. Then we rotate the data set around the y-axis so that this line is parallel to the z-axis.

Now we disregard from the z-components of the points, i.e., consider the points as projected onto the x,y-plane. Obviously, the two support vectors from V_+ will be projected on the same point, thus resulting in a situation with three support vectors in the x,y-plane; two from V_- and one (merged) from V_+. Having projected the support vectors on the x,y-plane,we use the same method of rotation as in the previous section, which does not affect the z-component of the points. Again we have produced a separating hyperspace with a larger margin.

For n> 3 dimensions we again cannot do anything if we only have two support vectors. In the case of 3, 4,..., n+1 support vectors we can, however, increase the margin, except in special cases. In order to simplify the discussion we can, by rotation, assume that the separating hyperplane is orthogonal to the x,y-plane.

If we have only three support vectors, we can directly project all data points on the x,y-plane by disregarding from the coordinates $x_3, \ldots, x_n$ for all points, perform the algorithm for $n = 2$, and then resubstitute the coordinates x_3 , $\ldots, x_n$ for all points, similarly to the case $n = 3$.

Now consider $|V_- \cup V_+| = k$ for $4 \leq k \leq n + 1$. We choose either V_- or V_+ which contains at least two points. We construct a line between two of the points in the set and rotate the data points so that this line becomes parallel with a base vector of dimension n, keeping the hyperplane orthogonal to the x,y-plane. We then disregard from the nth coordinate, thus projecting orthogonally the points from $\mathbb{R}^n$ to $\mathbb{R}^{n-1}$ and the separating hyperplane to a hyperplane with n-2 dimensions in $\mathbb{R}^{n-1}$. Since two support vectors in the n-dimensional space are mapped to the same point in the (n-1)-dimensional space, there are now n support vectors in the (n-1)-dimensional space. This procedure can be re-

peated until we reach three support vectors, where we use the method described in the previous section.

This produces a new data set with a separating hyperplane that has a larger margin, in general. If this hyperplane is not a maximum margin surface, the procedure can be repeated on the new data set to increase the margin further until the margin is as close to $M(S)$ as desired.

In the end we may perform the inverses of all transformations in order to regain the initial data set with a separating surface that consists of folded hyperplanes, which has a larger margin except in special cases.

Based on the discussion above the algorithm for the general case with n dimensions is given below.

Algorithm: Hyperplane Folding for the General Case

1: Run the standard SVM algorithm on the dataset $S \subset \mathbb{R}^n$ ($n > 2$).

2: **if** $k = 2$ **then** terminate [2]

3: **if** the determined number of hyperplane folding iterations is reached **then** terminate

4: Rotate S so that all support vectors have value zero in dimensions higher than or equal to k.

5: Temporarily remove dimensions $\geq k$ from all points in S.

If $k = n + 1$, no dimensions are removed.

If $k = n$, one dimension is removed, and so on.

6: **if** $k = 3$ **then goto** step 12

7: Select two support vectors v_1 and v_2 from the same class (V_+ or V_-).

8: Rotate S so that the values in dimensions 1 to $k-1$ are the same for v_1 and v_2.

9: Remove temporarily dimension k from all points in S. [3]

10: $k = k - 1$

11: **goto** step 6

12: Run the 2-dimensional algorithm presented in Section 3 for one iteration.

13: Expand the dimensions back one by one in reverse order and do the inverse rotations.

14: **goto** step 1

[2] k denotes the number of support vectors, i.e., $k = |V_- \cup V_+|$.

[3] This means that v_1 and v_2 are now mapped to the same support vector.

Regarding the computational complexity of the general case hyperplane folding algorithm we have the following. Initially, we must run the standard SVM algorithm on the considered dataset, which implies $O(\max(m,n)\min(m,n)^2)$ complexity according to (Chapelle, 2007), where m is the number of instances and n is the number of dimensions (attributes). Steps 6 to 11 in the algorithm is a loop with $n-2$ iterations. In each iteration we rotate the dataset. One data point rotation requires n^2 multiplications. This means that the computational complexity of this part is $O(mn^3)$. In step 12 we run the 2-dimensional algorithm, which has complexity $O(m)$. In step 13 we rotate back in reverse order, which has complexity $O(mn^3)$. This means that the computational complexity for one fold operation is $O(\max(m,n)\min(m,n)^2) + O(mn^3) + O(m) + O(mn^3)$, i.e. it can be simplified to $O(\max(m,n)\min(m,n)^2) + O(mn^3)$. If $m > n$, we have $O(mn^2) + O(mn^3) = O(mn^3)$. If $n > m$, we have $O(m^2n) + O(mn^3) = O(mn^3)$, i.e., the total computational complexity for one fold operation is $O(mn^3)$.

5. Initial Evaluation and Discussion

In order to get a better understanding of how the proposed hyperplane folding method works, we have conducted an experiment for the two-dimensional case. We implemented our algorithm in Python 2.7 using the Scikit-learn library[4], the Numpy library[5], and the Pandas library[6].

5.1. Data

We have generated n circles with synthetic data points (n = 4, 5, 6): $\lceil n/2 \rceil$ circles from S_+ and $\lfloor n/2 \rfloor$ circles from S_-, respectively. The circles are numbered 1 to n: odd numbered circles contain points from S_+, and even numbered circles contain points from S_-.

5.2. Experiment

Initially, we have studied how the margin in the SVM is influenced by our hyperplane folding method. For this purpose we have generated the following data set. Each circle i ($i = 1, 2, \ldots, 6$) is centered at (100+100i, 100+101(i mod 2)), e.g., circle number 3 is centered at (400, 201). The radius of each circle is set to 50 and 100 data points at random locations within each circle are generated. The generated data set is linearly separable, since the distance between in the y-dimension between S_+-circles and S_--circles is 101 and in addition, the radius of each circle is 50. This is

[4] http://scikit-learn.org/

[5] http://www.numpy.org/

[6] http://pandas.pydata.org/

Table 1. Margin and accuracy for the different SVMs. Each values is the average of 9 executions.

SVM number	4 circles		5 circles		6 circles	
	Margin	Accuracy	Margin	Accuracy	Margin	Accuracy
0	13.2	95.2%	7.8	94.9%	9.3	95.1%
1	15.9	96.8%	11.0	95.7%	11.3	95.5%
2	27.3	98.0%	12.3	96.4%	13.4	96.0%
3	33.8	98.4%	16.9	97.0%	14.5	96.2%

dataset 0. SVM 0 is obtained based on dataset 0. Then the hyperplane folding algorithm defined in Section 3 is used to create dataset i ($i = 1, 2, 3$) and the corresponding SVM i. Namely, dataset i ($i = 1, 2, 3$) is obtained by running our 2D algorithm on dataset i - 1. The corresponding SVM i ($i = 1, 2, 3$) is obtained based on dataset i.

We calculate the margin for each SVM i ($i = 0, 1, 2, 3$) (see Table 1).

The proposed method includes rotations of parts of the dataset. This means that we also need to rotate an unknown data point if we want to use the SVM for classification of unknown points. In the 2D case it is simply necessary to remember the line used for splitting the data points into two parts and the rotation angle α. In order to classify an unknown data point we first check if the data point belongs to the part that was rotated in the first iteration. In that case we rotate the unknown data point with the angle of the first rotation. We then check if the unknown data (that now may have been rotated) is in the part that was rotated in the second iteration. In that case we rotate the unknown data point with the angle of the second rotation, and so on. When all rotations are done we use the final SVM, i.e., the SVM we get after the 2D algorithm has terminated, to classify the data point in the usual way.

At the second phase of our evaluation, we have studied how the hyperplane folding method affects the classification accuracy of the SVM. For this purpose we have generated another data set. Each circle i ($i = 1, 2, \ldots, 6$) is centered at (100i, 100+101(i mod 2)). Then we set the radius of each circle to 75 and generate 1000 data points at random locations within each circle. This is test set 0. It is clear that some data points in test set 0 will be misclassified by SVM 0, since the radius for each circle is now 75. Table 1 shows the accuracy when classifying test set 0 using SVM 0, e.g., it is 95.2% for the case with 4 circles.

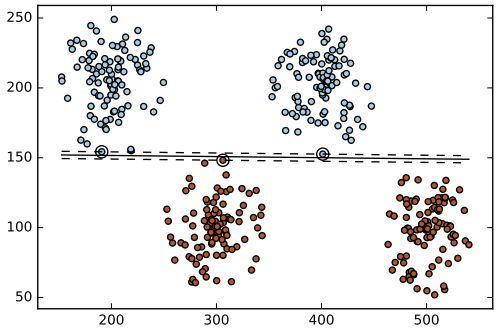
Then the data points in test set 0 are rotated in the same way as it was done when dataset 1 was obtained from dataset 0. This generates test set 1. The accuracy when classifying test set 1 using SVM 1 can be seen in Table 1, e.g., it is 96.8% for the case with 4 circles. We then continue rotating in the same fashion in order to obtain test set i ($i = 2, 3$) from test set i - 1. The corresponding accuracies when classifying test set 2 (or test set 3) using SVM 2 (or SVM 3) are given in Table 1. For instance, they are 98.0% and 98.4% for the case with 4 circles, respectively.

The obtained results (see Table 1) show that our method increases the margin significantly. The effect is most visible for the case with 4 circles. When we only have four of circles we will quickly get close to the largest possible margin, since we only have two circles of each class and we will in 2-3 iterations "fold away" one circle on each side of the separating hyperplane. This effect is illustrated graphically on Figures 6 and 7. Figures 6(a) and 7(a) show SVM 0 and SVM 2 for one of the nine tests for the case with four circles. Figures 6(b) and 7(b) show the corresponding test sets, i.e., test set 0 and test set 2. The angle and place of the two rotations can be clearly seen in test set 2. These figures demonstrate that already after two iterations we have folded the hyperplane so that the blue circle on the left side and the red circle on the right side are moved away from the separating hyperplane. When we have 5 or 6 circles the separating hyperplane needs to be folded more times to reach the same effect. This is the reason why the margin increases more quickly for the 4 circles case compared to the cases with more circles.

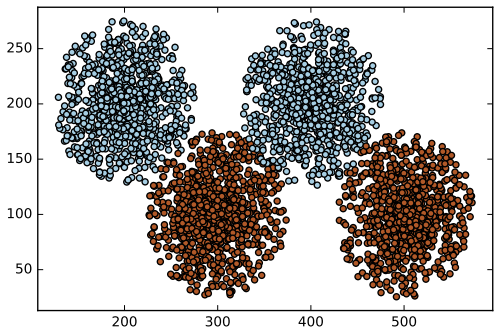
Table 1 also shows that the accuracy, i.e., the number of correctly classified data points in the test set divided with the total number of data points in the test set, also increases with our method. The reason for this is that a larger margin reduces the risk for misclassifications. The improvement in accuracy is again highest for the case with 4 circles. As it was discussed above the reason for this is that the margin increases fastest for that case.

5.3. Discussion

In the previous subsection we have discussed an experiment for the two-dimensional case performed for initial evaluation of the proposed hyperplane folding method. In the general case with n dimensions, we could do the same thing as in the 2D case, i.e., we start by deciding if the unknown data point is in the part of the n-dimensional space that was rotated in the first iteration. After that we deter-

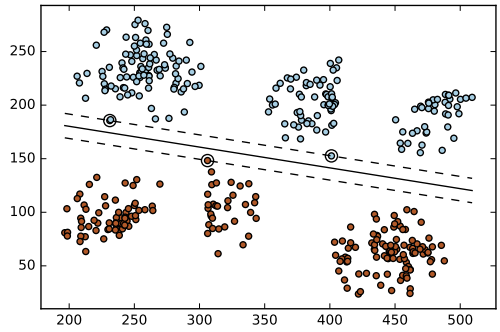


(a) SVM 0

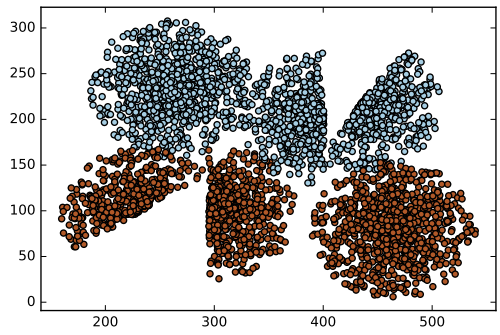


(b) Test set 0

Figure 6. Example of an SVM and test set for the case with four circles before any hyperplane folding iteration.



(a) SVM 2



(b) Test 2

Figure 7. Example of an SVM and test set for the case with four circles after two hyperplane folding iterations.

mine if the unknown data point (that now may have been rotated) is in the part of the n-dimensional space that was rotated in the second iteration, and so on. When all rotations are done we use the final SVM, i.e., the SVM we get after the algorithm in Section 3.3 has terminated, to classify the data point in the usual way.

The basic technique in hyperplane folding is to split the dataset into two parts and rotate one of the parts, and then repeat the same procedure again. It is clear that this technique could also be used for cases where the classes are not linearly separable, i.e., for soft margin SVMs. One way to do this is to move data points from one class in the direction of the normal of the separating hyperplane in the soft margin SVM until the dataset becomes linearly separable; follow our algorithm up to the point where we split the dataset; move the data points back to their original position; and then run the (soft margin) SVM on each part; rotate one of the parts; and finally merge the two parts (there could be other ways of implementing a soft margin version of hyperplane folding). Clearly, hyperplane folding could turn a dataset that is not linearly separable into a linearly separable dataset. Test set 2 in Figure 7(b) is almost linearly separable, and it is clear that if we had a radius of 60 instead of 75 the test set would be linearly separable after two iterations, whereas it is clearly not separable before we have done any iteration.

In the n-dimensional case we do n-2 dimension reductions before we reach the x,y-plane where the actual increase of the margin is obtained. Depending on the order that we reduce the dimensions we will end up with different x,y-planes. Some reduction orders will probably lead to more significant increases of the margin than other reduction orders. Finding the optimal reduction order is still an open research question. Moreover, the statistical bounds associated with support vector machines need to be adjusted when doing multiple folds in higher dimensions. This is a topic for further investigations.

Folding the hyperplane many times will result in a larger

margin. However, excessive folding could probably lead to overfitting. It is clear that the time for obtaining the SVM grows with the number of folds. One could also expect that the time required to classify an unknown data point will increase with many folds (even if there could be techniques to limit this overhead). This means that there is a trade-off between a large margin on the one hand and the risk of overfitting and the execution time overhead on the other hand. The margin is increasing in the number of iterations of hyperplane folding, and the algorithm can be stopped at any point. This means that we can balance the advantages and disadvantages of hyperfolding by selecting an appropriate number of iterations, e.g., we can simply stop when we do not want to spend more time on hyperplane folding or when the problem with overfitting becomes an issue.

We have assumed that for an n-dimensional dataset there can be at most $n+1$ support vectors. If we have limited resolution, there could be more than $n+1$ support vectors, and in such cases we need to do a small variation of the algorithm. The main idea in this case is to select the primary support vector, i.e., the support vector at which we split, so that the primary support vector is the only support vector from its class in one of the parts and so that that part also contains at least one support vector from the other class. It is clear that one can always do such a split when we have more than $n+1$ support vectors.

6. Conclusions

We have defined a method called hyperplane folding for support vector machines. The defined method increases the margin in the SVM. In addition, this method is easy to implement since it is based on simple mathematical operations, i.e., splitting and rotating a set of data points in n-dimensions, and the method then uses existing SVM implementations on the different parts of the dataset.

We have proposed hyperplane folding algorithms for the 2-dimensional and general n-dimensional cases, respectively. An initial evaluation of the algorithm for 2-dimensional case has been conducted. The obtained results have shown that the margin indeed increases and in addition, this improves the classification accuracy of the SVM.

A similar approach can be defined for the soft margin case, i.e., the case when the dataset is not linearly separable. The hyperplane folding method is incremental in the sense that the margin is improved in each iteration. Therefore we can adapt the number of folds to balance the risk of overfitting and the execution time on the one hand, and the size of the margin on the other hand.

Additional studies are needed before the potential of hyperplane folding can be fully understood. For future work, we also aim to pursue further evaluation and validation of the proposed hyperplane folding method on richer data.

References

Bartlett, P. L. (1998). The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, *44*, 525–536.

Chapelle, O. (2007). Training a support vector machine in the primal. *Neural Computation*, *19*, 1155–1178.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273–297.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The annals of statistics*, 1171–1220.

Huang, X., Mehrkanoon, S., & Suykens, J. A. (2013). Support vector machines with piecewise linear feature mapping. *Neurocomputing*, *117*, 118–127.

Li, Y., Leng, Q., Fu, Y., & Li, H. (2014). Growing construction of conlitron and multiconlitron. *Knowledge-Based Systems*, *65*, 12–20.

Lin, S., & Costello, D. J. (1983). *Error control coding: Fundamentals and applications*. Prentice-Hall.

Shawe-Taylor, J., Bartlett, P. L., Williamson, R. C., & Anthony, M. (1998). Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, *44*, 1926–1940.

Shawe-Taylor, J., & Cristianini, N. (2000). Margin distribution and soft margin. In *Advances in large margin classifiers*, 349–358. MIT Press.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer.

von Luxburg, U., Bousquet, O., & Schölkopf, B. (2004). A compression approach to support vector model selection. *Journal of Machine Learning Research*, *5*, 293–323.

Wang, S., & Sun, X. (2005). Generalization of hinging hyperplanes. *IEEE Transactions on Information Theory*, *51*, 4425–4431.

Yujian, L., Bo, L., Xinwu, Y., Yaozong, F., & Houjun, L. (2011). Multiconlitron: A general piecewise linear classifier. *IEEE Transactions on Neural Networks*, *22*, 276–289.

Towards Optimizing the Public Library: Indoor Localization in Semi-Open Spaces and Beyond

Martijn van Otterlo m.van.otterlo@vu.nl
Martin Warnaar m.p.c.warnaar@student.vu.nl
Economics & Business Administration, Computer Science, Vrije Universiteit Amsterdam, The Netherlands

Abstract

We report on the BLIIPS project which aims at the digitalization and optimization of physical, public libraries through the use of artificial intelligence combined with sensor technology. As a first step we introduce FLib, a localization application, and additional developments for interaction with physical books. The contributions of this paper are the introduction of the public library as an interesting testbed for smart technologies, a novel localization application with an experimental evaluation, and a compact research agenda for smart libraries.

1. Introduction

Under names such as *the extensible library* and *library* 3.0., the *public library* is changing (Allison, 2013). In our digital society, a constant stream of innovations and artificially intelligent algorithms turns every possible (*physical*) interaction in society into *data* from which *algorithms* can create *value* in some way (Zheng et al., 2014). One could assume that *public libraries*, with their *physical books*, would become obsolete now information and knowledge rapidly become digital, and huge tech-companies take over. For example, Wikipedia gives us encyclopedial knowledge, Google Books has many digitalized books, and Mendeley archives bibliographic information.

The *future of the library* is a much-debated topic which indicates that libraries have always been changing because of new technology in writing, printing, archiving and distributing. More than fifty years ago the visionary J.C.R. Licklider ((1965):33) wrote: *"By the year 2000, information and knowledge may be as important as mobility. We are assuming that the average man of that year may make a capital investment in an "intermedium" or console his intellectual Ford or Cadillac comparable to the investment he now makes in an automobile, or that he will rent one from a public utility that handles information processing as Consolidated Edison handles electric power."* We can see the modern *smartphone* or laptop being substituted in this quotation and indeed much of our contemporary information consumption is done through these devices. But despite all electronic access to information, the public library is still *the physical place* to go to (Palfrey, 2015) for *physical* books (Baron, 2015) and access to internet, but also things like 21st-century skill building and group activities. Public libraries are innovating in the direction of building communities of interest more or less connected to information: *from collection to connection* (see also (Palfrey, 2015)).

Here we report on project BLIIPS in which smartphones (and other technologies) are used for an orthogonal purpose: to *digitalize* physical interactions in the physical library to obtain insight in how physical public libraries are used and how their services can be optimized. In particular we introduce FLib, a *localization* application which uses machine learning to capture the signal landscape of both WiFi and Bluetooth beacon sensors for localization and navigation in a physical, multi-floor library building. The overall goal of BLIIPS is *to optimize the public library*, which can be about the layout of the physical space, the content and distribution of the book collection over this space and the visiting patterns of users. The public library is an excellent, semantically-rich, and much underexplored, environment for potential artificial intelligence research such as activity recognition, internet-of-things, optimization (logistics, space, services), and recommender systems.

This paper and contributions are structured as follows. In Section 2 we describe a much underexplored problem domain for artificial intelligence: the physical pub-

Appearing in *Proceedings of Benelearn 2017*.

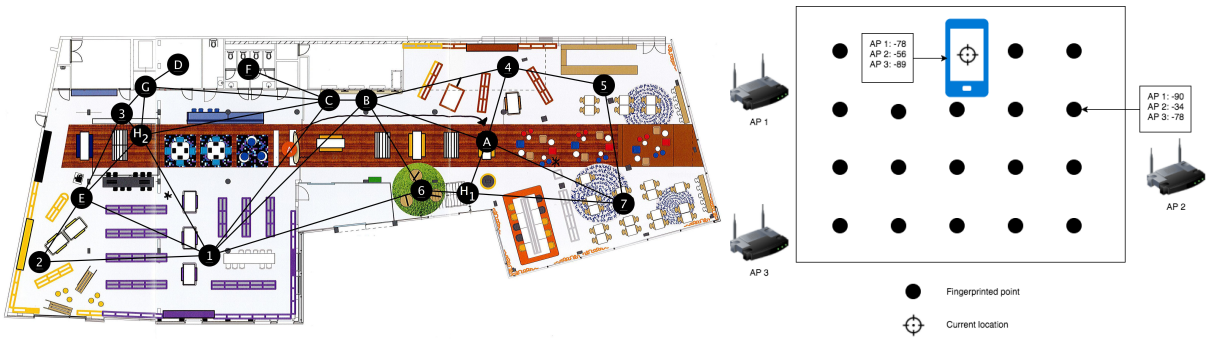


Figure 1. **a**) Alkmaar library ground floor map with an example topology of typical patron walking paths between meaningful locations. The purple area surrounding location (1) is "culture and literature" and the area at (2) features "young adult fiction". Both areas feature rectangular bookshelves and book-tables. Other locations are the self-service machines at (C) and (B), a staircase to the second floor (H1), a seating area (7) and the main desk at (A). The space is an open environment but furniture does restrict possible moving patterns, and in addition there are some small rooms at the top (e.g. behind C and B). **b**) Fingerprint positioning. At the *current* location (of the mobile phone) signals of all three APs are received. At the location on the right this holds too but AP2s signal is much stronger (-34) because it is closer.

lic library. In Section 3 we extensively introduce our FLIB localization application based on WiFi and beacon technology. In Section 4 we additionally mention methods for *book interaction* and conclude with a research agenda for further research in the public library.

2. The BLIIPS project

Companies and governments are looking for ways to utilize their existing data and capture new opportunities to develop initiatives around *data*. In the Dutch municipality of Alkmaar, such activities are aggregated through so-called *triple-helix cooperations* in which (local) governments, companies and knowledge institutions collaborate (van Otterlo & Feldberg, 2016). The Alkmaar public library, partially funded by the local government, works together with the Vrije Universiteit Amsterdam on a data-oriented project called BLIIPS (van Otterlo, 2016b). Its goal is to utilize data to *optimize the public library* in various ways. However, whereas most data-oriented projects are about already digitalized aspects, BLIIPS targets the *physical* aspects of the public library and seeks to *digitalize* them with the use of *new sensor technology*. The overall goal is to gain insight into the physical behavior of *patrons* (i.e. library "customers") in the physical library and how to optimize services, for example book borrowing.

2.1. Public libraries: Physical vs. Digital

The main library of Alkmaar is part of the *Kennemerwaard* group (out of about 150 groups) with 14 locations. Nationwide [1], in 2014 more than 22 percent of all Dutch people was member of a library, more than 63 million visits were paid to a library, and almost 80 million books were borrowed. However, libraries know very little about their patron's behavior. In fact, the only behavior visible in *data* are the books that were checked out. How they use the physical space, how they browse the book collection, which books are being looked at; for this no (real-time) data is available, but could be highly relevant for managing the physical library building, its services and its collection. Libraries do have a long history of *measuring*, *observing* and *evaluating* but typically through labor-intensive *surveys* and *observational studies* (see (Edwards, 2009; Allison, 2013; van Otterlo, 2016b) for pointers).

The BLIIPS project represents a first step towards *the intelligent library* in which this data is collected and analyzed in real time, but also in which the *physical* environment can provide to patrons "Google-like" services we are accustomed [2] to in the digital world. For example, if all interactions are digitalized, a smartphone could provide *location-based*, *personalized* recommendations to physical books in the patron's surrounding area, based on a user query and data about the library, the patron, and additional sources.

[1] `http://www.debibliotheken.nl/de-branche/stelsel/kengetallen-bibliotheken/`

[2] Related, but used in an orthogal way, van Otterlo ((2016a)) uses the concept of *libraryness* as a metaphor to understand modern profiling and experimentation algorithms in the context of privacy and surveillance.

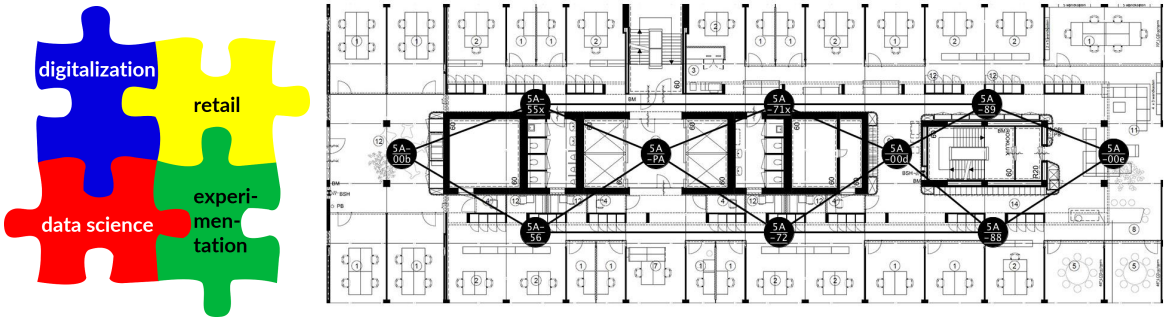


Figure 2. **a)** Four key developments in BLIIPS. **b)** Testing ground VU: an open office space at the 5th floor of the main building of the Vrije Universiteit Amsterdam. The topological graph shown depicts the transition structure of the hallway connecting all surrounding spaces of the floor. Many of the room-like structures are part of the open space, others are separate rooms with a door.

2.2. Towards Library Experimentation

BLIIPS builds upon four interlocking developments, see Figure 2**a** (van Otterlo, 2016b). The first puzzle piece is **digitalization**: making physical interaction digital through sensors (and algorithms). The second piece connects to **retail**: the use of smart technology in physical *stores*, such as the recent Amazon Go Store[3]. The Alkmaar library has adopted a *retail strategy* in which the layout and collection, unlike traditional libraries with many bookshelves, are more like a *store*: lots of visible book covers, tables with intuitive *themes* and easy categorizations that deviate much from traditional classification systems. A retail view on the problem appeals to *customer relations programmes*, *marketing* concepts and so-called *customer journeys*. The third piece concerns advances in *data science*, especially developments in machine learning and the availability of tools. The fourth piece of the puzzle, **experimentation and optimization**, is most important for our long-term goals. The BLIIPS acronym stands for *Books and Libraries: Intelligence and Interaction through Puzzle- and Skinnerboxes* in which the latter two elements denote physical devices used by psychologists in the last century for *behavioral engineering*. The aim of BLIIPS is to influence, or even control, behaviors and processes in the library in order to optimize particular goals, such as increasing the number of book checkouts. Such *digital Skinnerboxes* are becoming a real possibility due to the combined effect of data, algorithms and the ubiquity of digital interactions (van Otterlo, 2014). The library though, is a perfect environment for *experimentation*, unlike many other domains. As Palfrey (2015, p213) (quoting Kari Lamsa) writes: *"Libraries are not so serious places. We should not be too afraid of mistakes. We are not hospitals. We cannot kill people here. We can make mistakes and nobody will die. We can try and test and try and test all the time."*

[3] `https://www.amazon.com/b?node=16008589011`

3. An Indoor Localization Application

Mapping patron activity in the *physical* library requires at least knowing *where* they are. For this, we describe design and implementation of the FLIB application. Localization is a well-studied problem (Shang et al., 2015; He & Chan, 2015), but the practical details of the environment, hardware and algorithms used can deliver varying results and so far, *localization is not solved* (Lymberopoulos et al., 2015). First we outline requirements and then we describe an interactive localization application (Warnaar, 2017).

3.1. Indoor Localization

Whereas for outdoor localization GPS is successful, *indoor* localization remains a challenge. GPS works by maintaining line of sight to satellites which is problematic inside concrete buildings. Several indoor positioning systems exist (Shang et al., 2015; He & Chan, 2015) none of which is currently considered as *the* standard. Sensor information such as magnetic field strength, received radio waves, and inertial information from gyroscopes and odometers can be used to determine location. Smartphones are equipped with an array of sensors; they are well suited as indoor positioning devices. Lymberopoulos *et al.* (2015) review the 2014 *Microsoft indoor localization competition* (featuring 24 academic teams): *"all systems exhibited large accuracy*[4] *variations accross different evaluation points which raises concerns about the stability/reliability of current indoor location technolo-*

[4] Typically average errors of a couple of meters.

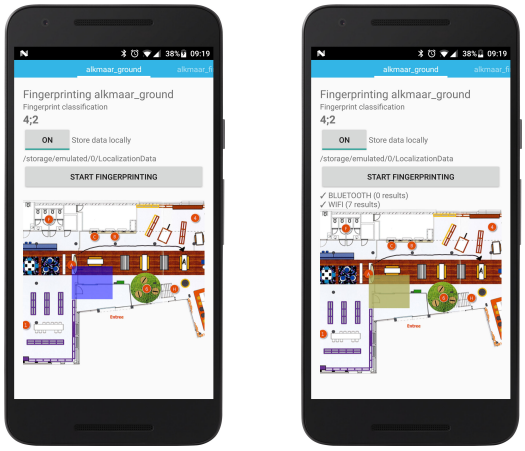


Figure 3. Android fingerprinting application

gies." Indoor spaces are often more complicated in terms of topology and spatial constraints: wireless signals suffer from multipath effect, scattering, and a non-line of sight propagation time, thereby reducing the localization accuracy. Due to the small scale, most applications require better accuracy than outdoors.

3.2. Library: Requirements and Solutions

Our target is the library of Alkmaar[5], a medium-sized city in the Netherlands. Its properties and the overarching BLIIPS project induce requirements for localization. First, the library consists of two floors (see Figures 1**a** and 5**a**) with mainly *semi-open space*. Unlike several *room-based* approaches, the library hardly contains constraints such as rooms/corridors. In terms of localization accuracy, we require (topical) area-based accuracy for effective navigation to library sections, and more accurate when technically feasible. Second, we want to leverage existing infrastructure as much as possible. Third, our solution needs to be amenable to *incremental accuracy improvement* without having to repeat all deployment steps. Fourth, obtained *data* in the deployment step should be reusable. Fifth, computational complexity should be low enough on smartphones. Sixth, smartphones should aid in obtaining required measurements for deployment. And last, the application needs to *engage the user* by visually showing the location and the patrons surroundings (or provide *navigation* in a later stage).

A common base of many solutions is *fingerprinting*: measuring the signal landscape such that localization amounts to matching currently sensed signals with the landscape. Shang *et al.* (2015) distinguish five main elements of a localisation solution: **(1) Sensors and hardware.** Localization depends on the interplay between sensor technology and devices such as smartphones. Sensors include wireless modules (e.g. WiFi, Bluetooth) and motion sensors (e.g. accelerometers, gyroscopes). Other well-known hardware are Zigbee and RFID chips. **(2) Measurements.** Localization depends on *what* is being measured from sensors. Most techniques use *received signal strength* (RSS) of a sensor. Derived measures such as *distances* and *angles* towards sensors are typically employed for *triangulation* approaches similar to GPS. An ideal signal should have two properties: *recognizability* and *stability*. **(3) Spatial contexts.** To aid localization, techniques such as *map matching*, *spatial models* (of behavior, but also of topological room connection structures) and *landmarks* can be used. These can be used in the localization process or as a top-level localization decision by employing the spatial context as a constraint. **(4) Bayesian filters.** Probabilistic approaches are effective for dealing with uncertainty in measurements. The most general formulation of Bayesian localization comes from Bayes' rule: $P(x|o) = \frac{P(o|x)P(x)}{p(o)}$ in which x is the location and o a set of measurements (the *observation*). A *sequence* of observations $o_1, o_2, \ldots, o_n$ is used to infer the sequence of locations $x_1, x_2, \ldots, x_n$, assuming there are (meaningful) (in)dependencies in the observations and locations. Such assumptions give rise to various probabilistic models that can be used for localization from noisy observations such as *(extended) Kalman filters* and *hidden Markov models*. All models have a bias wrt. choice of distributions used for e.g. *sensor models* $P(o|x)$ and how to do inference and learning. In prior experiments we employed a particular *Monte Carlo based probabilistic model* for localization, the *particle filter*, in our VU environment (see Figure 2**b**). A particle filter keeps multiple hypotheses (particles) of the location. Each time a patron moves, the particles are (probabilistically) moved based on a **motion model** to predict the next location. After sensing particles are resampled based on recursive Bayesian estimation using a **sensor model** that correlates the sensed data with the predicted state. Eventually the particles will converge on the true position. We concluded that obtaining accurate motion and sensor models was not feasible in this stage. A second bottleneck was the computational complexity on the phone when even a moderate number of particles and iterations were used. **(5) Hybrid localization.** A combination of techniques can improve the accuracy. These include multimodal fingerprinting, triangulation fusing multiple measurements, methods combining wireless positioning with pedestrian dead reckoning, and cooperative localization.

[5] `http://alkmaar.bibliotheekkennemerwaard.nl/`

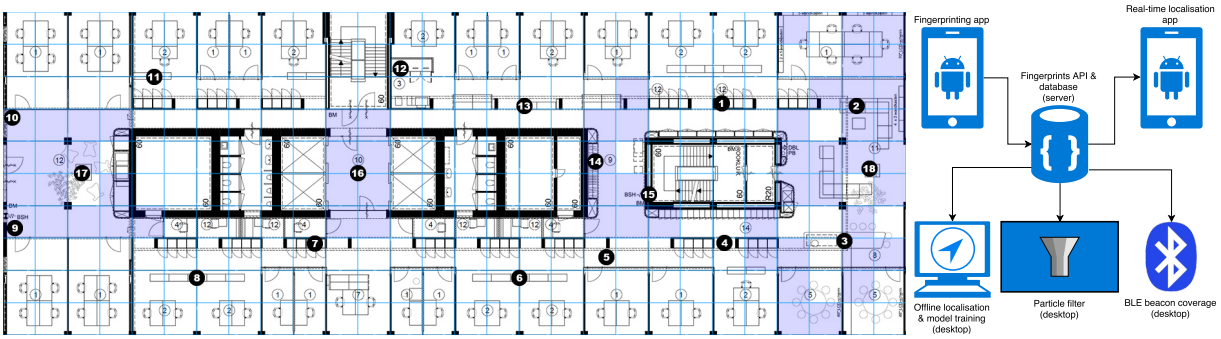


Figure 4. **a**) VU testing floor: 28×10 grid overlay and beacon positions. Coverage shown for beacons 3, 9, 14, and 16. **b**) FLIB :Software components overview.

Each localization technique has drawbacks when considering accuracy, cost, coverage, and complexity. None of them can be suitable to all scenarios. Based on our requirements formulated above we choose a (multimodal) *fingerprinting* solution in which we use smartphones both for measuring the signal space and for patron localization. Fingerprinting is accurate enough for our purpose, does not pose any assumptions on knowledge about where signals come from nor on the modeling of the domain (e.g. sensor models), and can be employed using the existing WiFi infrastructure which we extend with *Bluetooth beacons*. Other requirements (like low computational complexity and local computation) are fullfilled by the choice of (simple) algorithms with few biases and interactive visualizations on the phone, and because fingerprinting supports reuse of data. We use simple topological graphs and grid-based decompositions of space tailored to the required localization precision.

3.3. Localization by Fingerprinting

Localization by fingerprinting is a widely employed technique (He & Chan, 2015). The general principle is depicted in Figure 1**b**. Each black dot is a **reference point**: a location in the space from which all received signals together form a **fingerprint**. In the picture two received signal sets are depicted for two different reference points. More formally, let R be the set of reference points and A be the set of APs. We denote a sensed signal with strength s from AP $a \in A$ as the tuple (a, s). Now, let $f = \{(a_1, s_2), (a_2, s_2), \ldots, (a_n, s_n)\}$ be the set of all signals sensed at a particular location, called a fingerprint *over the set* A. A reference point can denote a point (x, y) in space (rendering R infinite), but usually is taken from a finite set of *regions*, *grid locations* (see Figure 5**a**) or *nodes* of an *abstract topological graph* such as in Figure 1**a**. A **fingerprint database** F_{DB} is a set of pairs (r, f) where $r \in R$ is a reference point and f a fingerprint over the set A.

In the **Offline training phase** we first **collect data**. Here a reference point r is (physically) visited to measure the signals available (f) at that location and to store (r, f) in the database. To increase the accuracy of F_{DB}, multiple fingerprints can be taken at the same location. Systematically all reference points $r \in R$ should be visited. When **building prediction models** the fingerprint database F_{DB} is used to obtain a generalizable mapping $M :: 2^{A \times \mathbb{R}} \to R$, i.e. a mapping from a set of signals (and their signal strengths) to a reference point in R. All samples $(r, f) \in F_{DB}$ represent a supervised learning problem from fingerprints (inputs) to reference points (outputs). In the **Online localization phase.** M is used for localization. Let the to-be-located-patron be in some unknown location l in the space, and let the set of current signals be $c = \{(a_1, s_2), (a_2, s_2), \ldots, (a_n, s_n)\}$. The predicted location of l is then $r = M(c)$.

The choice for fingerprinting naturally induces a *supervised machine learning* setting in which the signal landscape over the space is the desired function to learn, and where the fingerprints are *samples* of that function. Intuitively, this determines the balance between $|R|$ and *sample complexity* (Wen et al., 2015). Fingerprinting is not prone to error drift such as often seen when using inertial sensors to determine step count and direction. Modelling signal decay over distance and through objects is also not required, as is the case for multilateration positioning. Another advantage is that the positions of APs do not need to be mapped. Disadvantages are that collecting fingerprints of the site is a tedious task (Shang et al., 2015) and that changes to the environment may require that (some) fingerprints need to be collected again.

3.4. Multimodal Fingerprinting: Beacons

One of the constraints is that the library building has only 8 different WiFI APs. Several other APs from surrounding buildings can be used but they are outside our control and less reliable. In contrast, our test VU environment (see Figure 2**b**) has many APs inside. To enrich the signal landscape, we employ so-called *Bluetooth low energy (BLE) beacons*. A beacon is a self-powered, small device that sends out a signal with a adjustable signal strength and frequency. Beacons are a recent addition to the *internet-of-things-landscape* (Ng & Wakenshaw, 2016) and most modern smartphones can detect them. For example, a museum can place a beacon at an art piece and when the visitor gets near the beacon, his smartphone can detect this and provide information about the object. Most work employs beacons for areas such as rooms and corridors (i.e. *region-based*). For example, LoCo (Cooper et al., 2016) is a fingerprinting system based on WiFi APs and BLE beacons which are mostly aligned with the room-like structure of an office. Such beacons act as noisy indicators for rooms. Such constraints are somewhat present in our VU environment, but not at the library. In a sub-project (Bulgaru, 2016) we tested region-based interpretations in the library with varying success due to the noisy nature of beacons.

Here, in our semi-open library space we opt for a more general approach; to employ beacons as *extra* signals for fingerprinting and to treat them similar to WiFi signals. Beacons are configured such that signals will be received throughout large portions of the library, just like the (much stronger) WiFi APs. Using this approach roughly 10 beacons per floor are effective. Consequently, in our model, the set A of all APs is extended with all beacons $\{b_1, \ldots, b_n\}$.

3.5. The FLib Localization Application

In this section we describe FLib, a smartphone application for localization purposes in a real-world environment. Figure 4**b** shows an overview of main (software) components of FLib. In subsequent sections we will review all parts. FLib is targeted at our testing ground at the university (see Figure 2**b**) and the library in Alkmaar (see Figures 1**a** and 5**a**).

3.5.1. Fingerprinting

The fingerprint database F_{DB} is filled by smartphone measurements, see Figure 3. In FLib the current position can be selected on an interactive map, after which the fingerprint is generated with a single button tap. Initial experiments in VU and Library employed a graph-like transition structure as in Figure 1**a** which was replaced by a more uniform grid layout as depicted in Figures 4**a** and 5**a**. When the user fingerprints a grid location, it gets highlighted to keep track of visited areas. The Estimote Android SDK is used to collect Bluetooth signals. WiFi RSSIs are collected using Android's `BroadcastReceiver`. The Estimote software uses an adaptable *scanning period* and a *pause* between scanning periods. If the first is too short, few or no beacons are detected, but if it is too long location-estimation lags behind (and: huge performance differences between smartphones exist).

3.5.2. Fingerprints Server

Measured fingerprints are uploaded to a server application, (implemented in PHP using Symfony running on Apache, using a MySQL database). Fingerprints are first locally stored on the phone and then sent to the server. The server's only function is to store fingerprints data: localisation runs locally on the phone.

3.5.3. Model Training

The data on the server F_{DB} is used for building a model mapping fingerprints to grid locations (reference points). We utilize two different machine learning algorithms: k-nearest-neighbors (k-NN) and multi-layer perceptrons (MLP), see (Flach, 2012).

The first model is a *lazy* learner; generalization and model building is not required, but instead F_{DB} is loaded on the smartphone and the algorithm finds the k most similar fingerprints in F_{DB} for the currently sensed signals. We use a *modified Euclidean distance* to compute a *similarity metric* between fingerprints. Given a fingerprint $f = \{(a_1^f, s_1^f), \ldots, (a_n^f, s_n^f)\} \in F_{DB}$ and the currently sensed signals, $c = \{(a_1^c, s_1^c), \ldots, (a_m^c, a_m^s)\}$, we compute the distance between c and f. Let $A^{fc} \subseteq A$ be access points measured in both c and f. We compute distance $d(c, f)$ as follows. For all sensed APs in A^{fc} we take the overall Euclidean distance between signal values. A *penalty* of 30 is added to the distance for each access point $a \in A$ that is *only* in f and *not* in c, or *only* in c and *not* in f. This empirically estimated value balances signals and missing values.

Our second model is an MLP, a standard neural network with one *hidden* layer of neurons, an *input* layer with $|A|$ neurons and an output layer with $|R|$ neurons. Reference points are taken as *classes* and each class ($r \in R$) is represented by a separate neuron. For the input layer we transform a sensed fingerprint $\{(a_1, s_1), \ldots, (a_m, s_m)\}$ with $m \leq |A|$ to a vector of length $|A|$ where each $a \in A$ has a fixed index in this vector and each value at that index is the sensed signal

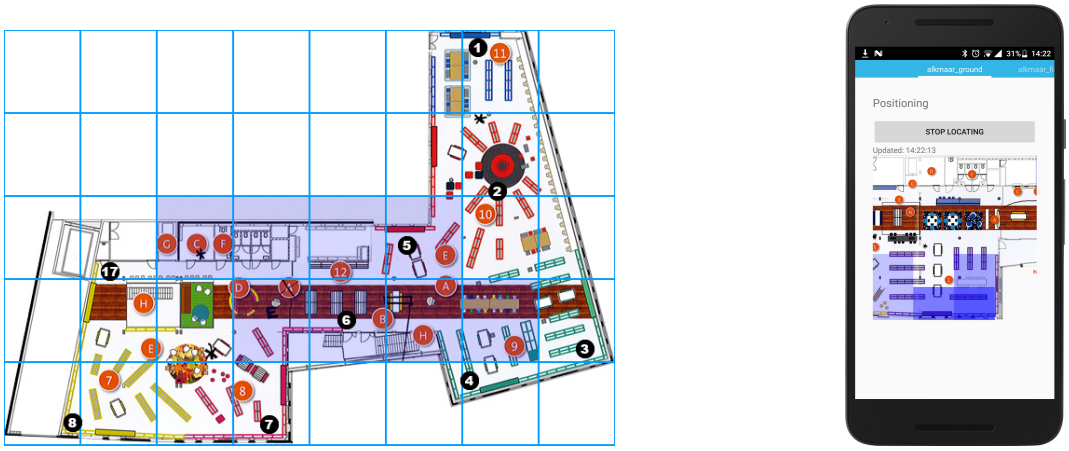


Figure 5. **a**) Alkmaar first floor (8 x 5 grid) with beacon positions (coverage shown for 6), **b**) **FLib** localization.

strength $s_i (i \in 1 \dots m)$. All other components of the input vector are 0. To construct an output vector for a fingerprint f (i.e. $(r, f) \in F_{DB}$) we use a binary vector of length $|R|$ with all zeros except at the index of the neuron representing r. Training an MLP amounts to computing the best set of *weights* in the network which can be accomplished using gradient-descent learning.

3.5.4. Real-time Localisation

Both models can be used to generate a *ranking* $\langle r_1, \dots, r_m \rangle$ of all reference points. k-NN naturally induces a ranking based on distances. MLPs however, yield a *normalized distribution* over the output nodes.

Instead of showing only the best prediction of location, FLib shows a more gradual visualization which highlights with shades of blue where the patron may be. To render the blue tiles as in Figure 5**b**, we calculate the *transparency* for the returned unique reference points. Let $R_{best} = \langle r_1, r_2, ..., r_n \rangle$ be the ranked locations where some $r \in R$ can occur multiple times. The first element gets score $|R_{best}|$, the second $|R_{best}| - 1$ and so on, and scores for the same r are summed. Scores are normalized and mapped onto $50 \dots 255$, inducing color value as a shade of blue.

3.6. Experiments and Outcomes

Experiments were conducted at two separate locations: part of the 5th floor of the main building at the VU university in the A-wing (VU) and the two publicly accessible floors at the Kennemerwaard Library at Alkmaar (LIBRARY). The LIBRARY ground floor is 55 m wide by 22 m in length, while the first floor is 54 m wide by 40 m in length. The VU testing floor is 57,4 m wide and 20,5 m in length. Estimote Proximity and BeaconInside beacons were both used. Transmision rate and power were (497ms, −16dBm) and (500ms, −3dBm) for Estimote and BeaconInside beacons respectively. Fingerprinting was done with different smartphones: OnePlus A3003 (3), LG H320 (Leon), and Huawei Y360-U61 (Y3). All access points in the vicinity are used for fingerprints collection to increase WiFi signal space. For VU, we have 396 unique AP addresses in the fingerprints collection, compared to 165 for LIBRARY. A Bluetooth scanning period of 2500 ms was used to balance delay and detection. RapidMiner was used to train MLPs (learning rate 0.3, momentum 0.2, normalized inputs) and inference in models runs on the phone.

3.6.1. Experimental Setup

First, we determine whether unlabelled walked trajectories can successfully be classified at VU. We use the graph model from Figure 2**b** and fill F_{DB} with fingerprints taken at each node position. Next, we walk several trajectories such as shown in Figure 6**a**, and store unlabelled fingerprints of multiple locations. Using 1-NN with the modified Euclidean function, the predicted sequences of reference points are compared to the truly walked paths.

Positioning performance over the grid at VU and LIBRARY (Figures 4**a** and 5**a**) is calculated by taking the mean hamming distance (H) between n true $(x, y) \in R$ and predicted reference points $(x', y') \in R$:

$$H(M) = \frac{1}{n} \sum_{i=1}^{n} |x_i - x'_i| + |y_i - y'_i| \qquad (1)$$

Fingerprints are collected with different phones, while fingerprints of walks were collected with a OnePlus 3

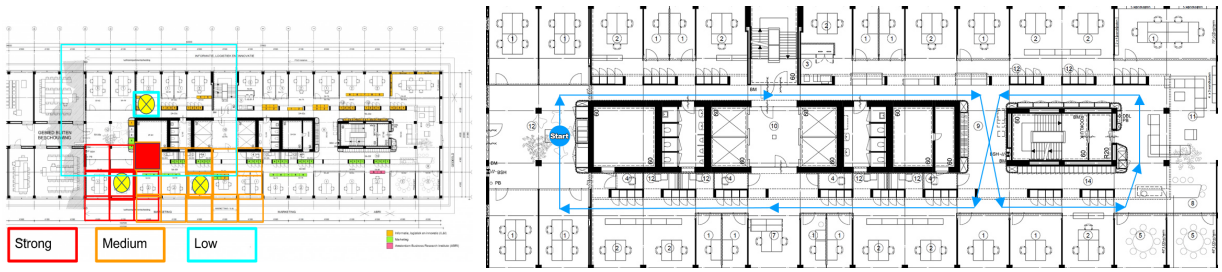


Figure 6. **a**) Picture from (Bulgaru, 2016). Here we employ beacons in our testing environment with exactly known beacon locations. An effective localization method is to score each grid location surrounding a detected beacon based on the detected signal strength. For example, all 9 grid locations around a detected *strong* beacon signal (red area) get a high value, whereas a much larger area around a detected *low* signal (blue) get a much lower value. This value scheme reflects varying confidence in beacon detections based on signal strength. The final predicted location is computed from a (weighted) combination of all grid position values, and forms a practical (and effective in the testing environment) localization algorithm. **b**) A sample walk in the VU environment (Walk 2).

only. Differences in performance are compared using only fingerprints of *OnePlus3*, *averaged* fingerprints, and *all* fingerprints. Best performance is expected when using fingerprints from the same phone as for the walks, since there are no sensor or configuration differences. In *all* fingerprints for the ground floor at LIBRARY, 745 records were collected, and 623 for the first floor. Averaging fingerprints data per phone per reference point was done to decrease computational complexity of k-NN, reducing $|f_{DB}|$ for the first floor from 623 to just 72. Computational efficiency is important because smartphones have limited battery time, and positioning delay is reduced.

3.6.2. RESULTS

First, we look at 2 VU walk example results:

True *W1* {5A-00b, 5A-55x, 5A-PA, 5A-71x, 5A-00d, 5A-89, 5A-00e, 5A-88, 5A-00d, 5A-72, 5A-56, 5A-PA, 5A-00b}
Predicted *W1* {5A-00b, 5A-55x, 5A-71x, 5A-00d, 5A-89, 5A-00e, 5A-00d, 5A-88, 5A-72, 5A-56, 5A-00b}
True *W2* {5A-00b, 5A-55x, 5A-PA, 5A-71x, 5A-00d, 5A-88, 5A-00e, 5A-89, 5A-00d, 5A-72, 5A-PA, 5A-56, 5A-00b}
Predicted *W2* {5A-89, 5A-00b, 5A-55x, 5A-71x, 5A-00d, 5A-00e, 5A-00d, 5A-89, 5A-00d, 5A-72, 5A-56, 5A-00b}

We see that predicted and true sequences are very similar, with the exception of some natural additional predicted neighboring locations. For VU, the positioning performance for our MLP and k-NN configurations are displayed in Figure 7**a**. In the best case, 2-NN, we have an average hamming distance error of 2.67 (1.65 in x and 1.02 in y). Each grid tile is 2.05 m in width and length, so we roughly (under)estimate the mean error with $\sqrt{(1.65 * 2.05)^2 + (1.02 * 2.05)^2} \approx 3.97$ m.

For LIBRARY, the accuracy of different configurations over averaged fingerprints (ground floor) is in Figure 7**b**. Figure 7**c** shows results for the first floor. Ground floor tiles cover 5.5 × 4 m. For the LIBRARY ground floor the best result (MLP, 50 hidden, 200 cycles) is a mean total hamming distance of 1.06: 0.65 for x and 0.41 for y and roughly (under)estimated error of $\sqrt{(5.5 * 0.65)^2 + (4 * 0.41)^2} \approx 3.92$ m. For the LIBRARY first floor, the same configuration yields the best result: 0.80, with an error $x = 0.35$ and $y = 0.45$. Each grid tile covers 6.75 × 8 m, giving an (under)estimated mean error of $\sqrt{(6.75 * 0.35)^2 + (8 * 0.45)^2} \approx 4.30$ m. These levels of indoor localisation performance suffice to detect a patron's region at LIBRARY, and can be used for several future applications. We have seen that using $k > 3$, positioning performance starts degrading, so only results of $\{1, 2, 3\}$-NN are reported.

3.7. Related Work

There is much related work in localization, e.g. (He & Chan, 2015; Shang et al., 2015; Lymberopoulos et al., 2015). Our major contribution is the library domain and its potential for library optimization; in terms of pure localization accuracy several systems may be better. However, the BLIIPS project's requirements on accuracy are less strong for the tasks we aim at. A relatively novel aspect is that we aim at semi-open spaces and do make different use of multimodal (i.e. with additional beacons) fingerprinting than in other systems such as (Cooper et al., 2016; Kriz et al., 2016). Direct comparison of empirical results is for this reason not feasible. In the past only very few such systems have been considered for (public) library settings (see (van Otterlo, 2016b) for pointers) and the results were very limited; here our contribution lies in a successful mix of previously known techniques in a library setting.

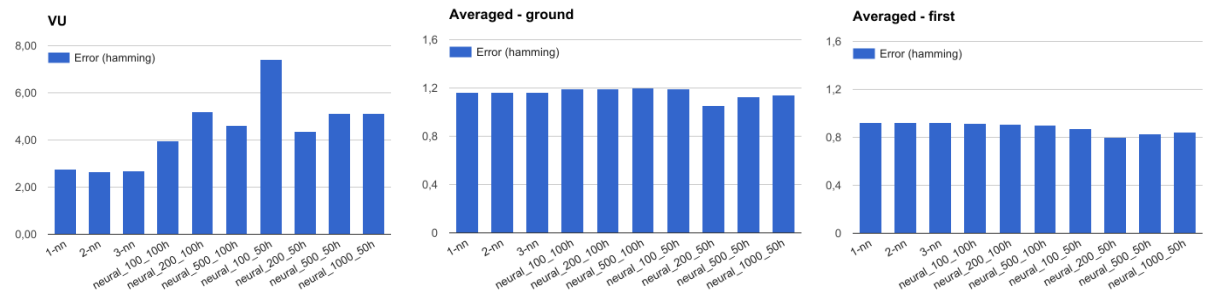


Figure 7. Mean positioning hamming distance errors: **a**) at the VU (28×10 grid), **b**) using averaged fingerprints at the LIBRARY ground floor **c**), using averaged fingerprints at the LIBRARY first floor.

4. Elements of a Research Agenda

The BLIIPS project has a main goal: to establish a *library experimentation facility* to *experiment* with ways to *optimize the public library services* by *influencing patrons*, for example by interacting with patrons through recommendations or by changing the layout of the library or its collection. The FLIB application represents a large step towards turning patron behaviors into actionable data. We envision many other challenges and opportunities for the intelligent library and briefly mention some directions. **(1) Fingerprinting/localization.** Current efforts go into extending and upgrading FLIB to increase accuracy and to incorporate more *spatial context* and other (semantic) constraints coming from the library. Other types of (deep) machine learning and especially (structured) probabilistic models are appealing. In addition, we want to investigate i) *collaborative* fingerprinting (using many devices), and ii) the optimization of the *choice* for, and *placement* of, sensors in the environment (e.g. (Shimosaka et al., 2016)). **(2) Digitalization.** In addition to our use of WiFi and beacons, a sub-project in BLIIPS targeted *interaction with physical books* (Jica, 2016), with *computer vision* and using RFID chips contained in each book (see Figure 8 for an example). Books can be looked up based on i) cover, ii) bar code, iii) RFID CHIP, iv) textual information (ISBN). Combined with localization (movement) data, this further completes the digitalization of library activity. However, we envision more ways to digitalize physical processes, using various new types of sensors, developments in networks (e.g. LoraWAN), existing technologies such as "smart shelves", augmented/virtual reality, and more. **(3) Activity recognition.** More detailed data of digitalized, physical *activities* can be distilled and used for *predictive models*. Traditional library research has analyzed data before, but the scale of current and potential behavioral data is virtually unlimited and

Figure 8. Example alternative technology tested in BLIIPS: processing the visual appearance of book spines to recognize pictograms denoting book types.

many interesting challenges await: can predictions be made *who* will do *which* activity, *for how long*, and with *what purpose*? A general *big data frame* can connect such data with demographics, geographical context, weather, trends on social media, and much more. **(4) The personalized library** In line with BLIIPS's philosophy of making the physical library more Google-like, *personalization* will be a big issue in the *data-driven* public library. Knowledge classification schemes, recommendations, advertisements and suggestions for activities could all be personalized based on data (e.g. books borrowed) combined with statistical predictions derived from many patrons. The physical setting enables *location-based* interventions such as personalized suggestions about interesting books in the local neighborhood. **(5) Optimizing all services.** *Prediction models* can enable *optimization* of processes by means of *experimentation*. For example, one can systematically change aspects of the library services using expectations and actually see the results in the data. Optimization requires *goals*. Potential library optimization goals that are largely unexplored are i) the number of books people borrow, ii) distribution of (types of) books in the collection, iii) the most efficient layout of the library, and iv) the conceptual arrangement of knowledge (classification schemes). One advantage of data-oriented approaches is that monitoring

and intervening can be done in *real time*. The advantage of sensor technology is that at some point one can *relax the physical order* of the library because, for example, books can be located individually, escaping the standard order of the shelf. Coming up with the right goals – together with the right hardware and algorithmic technology – that are aligned with the many functions of the public library, is most challenging. **(6) Privacy.** More data means more risks for *privacy* in general. Libraries already collect data about their patrons, but this will increase quickly. Challenges are basic data privacy and security. However, a more hidden form is *intellectual privacy* (see (van Otterlo, 2016a)). Personalized interventions in library services based on information about borrowing history can have *transformative effects* on the *autonomy* of a patron in thinking and deciding. Consequences of data-driven strategies in libraries are underexplored (but see (van Otterlo, 2016b)) and need more study.

5. Conclusions

In this paper we have introduced the public library as an interesting domain for innovation with artificial intelligence. In the context of project BLIIPS we have introduced the FLib localization application as a first step towards patron activity monitoring, and have briefly touched upon additional results related to book interaction. Many potential future work directions on BLIIPS and FLib exist and were outlined in the research agenda in the previous section.

Acknowledgments

The first author acknowledges support from the Amsterdam academic alliance (AAA) on data science, and we thank Stichting Leenrecht for financial support. We thank the people from the Alkmaar library for their kind support.

References

Allison, D. A. (2013). *The patron-driven library: A practical guide for managing collections and services in the digital age*. Chandos Inf. Prof. Series.

Baron, N. S. (2015). *Words onscreen: The fate of reading in a digital world*. Oxford University Press.

Bulgaru, A. (2016). Indoor localisation using bluetooth low energy beacons. Bachelor thesis, Vrije Universiteit Amsterdam, The Netherlands.

Cooper, M., Biehl, J., Filby, G., & Kratz, S. (2016). LoCo: boosting for indoor location classification combining WiFi and BLE. *Personal and Ubiquitous Computing*, *20*, 83–96.

Edwards, B. (2009). *Libraries and learning resource centres*. Architectural Press (Elsevier). 2nd edition.

Flach, P. (2012). *Machine learning*. Cambridge University Press.

He, S., & Chan, G. (2015). Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Comm. Surveys & Tutorials*, *18*.

Jica, R. (2016). Digital interactions with physical library books. Bachelor thesis, Vrije Universiteit Amsterdam, The Netherlands.

Kriz, P., Maly, F., & Kozel, T. (2016). Improving indoor localization using bluetooth low energy beacons. *Mobile Information Systems*, *2016*.

Licklider, J. (1965). *Libraries of the future*. Cambridge Massachusetts: MIT Press.

Lymberopoulos, D., Liu, J., Yang, X., Choudhury, R. R., Handziski, V., & Sen, S. (2015). A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. *Proceedings of the 14th International Conference on Information Processing in Sensor Networks* (pp. 178–189). New York, NY, USA: ACM.

Ng, I. C., & Wakenshaw, S. Y. (2016). The internet-of-things: Review and research directions. *International Journal of Research in Marketing*, *34*, 3–21.

Palfrey, J. (2015). *Bibliotech*. Basic Books.

Shang, J., Hu, X., Gu, F., Wang, D., & Yu, S. (2015). Improvement schemes for indoor mobile location estimation: A survey. *Math. Probl. in Engineering*.

Shimosaka, M., Saisho, O., Sunakawa, T., Koyasu, H., Maeda, K., & Kawajiri, R. (2016). ZigBee based wireless indoor localization with sensor placement optimization towards practical home sensing. *Advanced Robotics*, *30*, 315–325.

van Otterlo, M. (2014). Automated experimentation in walden 3.0. *Surveillance & society*, *12*, 255–272.

van Otterlo, M. (2016a). The libraryness of calculative devices. In L. Amoore and V. Piotukh (Eds.), *Algorithmic life: Calculative devices in the age of big data*, chapter 2, 35–54. Routledge.

van Otterlo, M. (2016b). Project BLIIPS: Making the physical public library more intelligent through artificial intelligence. *Qualitative and Quantitative Methods in Libraries (QQML)*, *5*, 287–300.

van Otterlo, M., & Feldberg, F. (2016). Van kaas naar big data: Data Science Alkmaar, het living lab van Noord-Holland noord. *Bestuurskunde*, 29–34.

Warnaar, M. (2017). Indoor localisation on smartphones using WiFi and bluetooth beacon signal strength. Master thesis, Vrije Universiteit Amsterdam.

Wen, Y., Tian, X., Wang, X., & Lu, S. (2015). Fundamental limits of RSS fingerprinting based indoor localization. *IEEE Conference on Computer Communications (INFOCOM)* (pp. 2479–2487).

Zheng, Y., Capra, L., Wolfson, O., & Yang, H. (2014). Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.*, *5*, 38:1–38:55.

Constraint-based measure for estimating overlap in clustering

Antoine ADAM ANTOINE.ADAM@CS.KULEUVEN.BE
Hendrik Blockeel HENDRIK.BLOCKEEL@CS.KULEUVEN.BE
KU Leuven, Department of Computer Science, Celestijnenlaan 200A, 3001 Leuven, Belgium

Keywords: meta-learning, constraints, clustering

Abstract

Different clustering algorithms have different strengths and weaknesses. Given a dataset and a clustering task, it is up to the user to choose the most suitable clustering algorithm. In this paper, we study to what extent this choice can be supported by a measure of overlap among clusters. We propose a concrete, efficiently computable constraint-based measure. We show that the measure is indeed informative: on the basis of this measure alone, one can make better decisions about which clustering algorithm to use. However, when combined with other features of the input dataset, such as dimensionality, it seems that the proposed measure does not provide useful additional information.

1. Introduction

For many types of machine learning tasks, such as supervised learning, clustering, and so on, a variety of methods is available. It is often difficult to say in advance which method will work best in a particular case; this depends on properties of the dataset, the target function, and the quality criteria one is interested in. The research field called meta-learning is concerned with devising automatic ways of determining the most suitable algorithm and parameter settings, given a particular dataset and possibly knowledge about the target function. Traditionally, meta-learning has mostly been studied in a classification setting. In this paper, however, we focus on clustering.

Clustering algorithms are no exception to the general rule that different learning algorithms make different assumptions about the input data and the target function to be approximated. For instance, some clustering algorithms implicitly assume that clusters are spherical; k-means is an example of that. Any clustering algorithm that tries to minimise the sum of squared Euclidean distances inside the clusters, implicitly makes that assumption. The assumption can be relaxed by rescaling the different dimensions or using a Mahalanobis distance; this can lead to elliptic clusters, but such clusters are still convex.

A different class of clustering algorithms does not assume convexity, but looks at local properties of the dataset, such as density of point or graph connectivity. Such methods can identify, for instance, moon-shaped clusters, which k-means cannot. Spectral clustering (von Luxburg, 2007) is an example of such approach.

Some clustering algorithms assume that the data have been sampled from a population that consists of a mix of different subpopulations, e.g., a mixture of Gaussians. EM is an example of such an approach (Dempster et al., 1977). A particular property of these approaches is that clusters may overlap. That is, even though each individual instance still belongs to one cluster, there are areas in the instance space where two (or more) Gaussian density functions substantially differ from zero, so that instance of both clusters may end up in this area.

In this paper, we hypothesise that the amount to which clusters may overlap is relevant for the choice of what clustering method to use. A measure, the Rvalue, has been proposed before that, given the ground truth regarding which instance belongs to which cluster, describes this overlap. Since clustering is unsupervised, this measure cannot be used in practice for deciding what clustering method to use. We therefore derive a new measure, CBO, which is based on must-link or cannot-link constraints on instance pairs. We show that the second measure correlates well with the first, making it a suitable proxy for selection the clustering

Preliminary work. Under review for Benelearn 2017. Do not distribute.

method. We show that this measure is indeed informative: on the basis of this measure alone, it is possible to select clustering algorithms such that, on average, better clusterings are obtained.

However, there are also negative results. Datasets can be described using other features than the measure defined here. It turns out that, when a dataset is described using a relatively small set of straightforward features (such as dimensionality), it is also possible to make an informed choice about what clustering method to use. What's more, if this set of straightforward features is extended with the overlap measure described here, this does not significantly improve the informativeness of the dataset description, in terms of which clustering method is optimal.

The conclusion from this is that, although the proposed measure is by itself an interesting feature, it seems to capture mostly information that is also contained in other, simpler features. This is a somewhat surprising result for which we currently have no explanation; further research is warranted.

This paper is the continuation of a previously published workshop paper (Adam & Blockeel, 2015). While following the same ideas, the CBO has been completely redefined. In addition, the number of datasets considered was increased from 14 to 42. While the correlation of the CBO with the overlapping has improved considerably, the promising results for the algorithm selection of that paper were somewhat reduced by adding those datasets.

The remainder of this paper is structured as follows. Section 2 discusses some related work on constraint-based clustering and meta-learning for clustering. Section 3 studies how the overlapping of clusters influences the performance of algorithms. Section 4 introduces CBO, which is intended to approximate the amount of overlap from constraints. Section 5 presents experimental results that compare algorithm selection based on CBO with algorithm selection using other features of the dataset. Section 6 presents our conclusions.

2. Related work

2.1. Constraint-based clustering

Clustering is the unsupervised learning task of identifying groups of similar instances in a dataset. Although these groups are initially unknown, some information can be available as to what the desired solution is. This information takes the form of constraints on the resulting clusters. These constraints can be provided to the clustering algorithm to guide the search towards a more desirable solution. We then talk about constraint-based, constrained, or semi-supervised clustering.

Constraints can be defined on different levels. On a cluster level, one can ask for clusters that are balanced in size, or that have a maximum diameter in space. On an instance level, one might know some partial labelling of the data. A well-used type of constraints are must-link and cannot-link constraints, also called equivalence constraints. These are pair-wise constraints which state that two instances must be or cannot be in the same cluster.

Multiple methods have been developed to use these constraints, some of which are mentioned below. A metric can be learnt that complies with the constraints (Bar-Hillel et al., 2005). The constraints can be used in the algorithm for the cluster assignment in a hard (Wagstaff et al., 2001) or soft way (Pelleg & Baras, 2007), (Ruiz et al., 2007), (Wang & Davidson, 2010). Some hybrid algorithms use constraints for both metric learning and clustering (Bilenko et al., 2004), (Hu et al., 2013). Other approaches include constraints in general solver methods like constraint programming (Duong et al., 2015) or integer linear programming (Babaki et al., 2014).

2.2. Algorithm selection for clustering

Little research has been conducted on algorithm selection for clustering. Existing methods usually predict the ranking of clustering algorithms (De Souto et al., 2008), (Soares et al., 2009), (Prudêncio et al., 2011) (Ferrari & de Castro, 2015). The meta-features used are unsupervised and/or domain-specific. None of these approaches are using constraints which removes the specificity that there is not only one single clustering for one dataset. To the best of our knowledge, the only meta-learning method for clustering involving constraints is (Van Craenendonck & Blockeel, 2016) which does not use features but simply selects the algorithm that satisfies the most constraints.

3. Rvalue

As already mentioned, we assume some algorithms can handle overlapping better than others. For example, figure 1 shows a toy dataset (on the left) where two Gaussians overlap in there centre, forming a cross. In that case, EM (in the middle) is capable of retrieving the correct clustering while spectral clustering (SC, on the right) cannot. This shows the relevance of overlapping as a meta-feature to select a clustering algorithm.

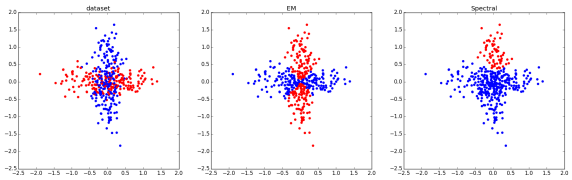


Figure 1. Toy example of the cross dataset.

The Rvalue (Oh, 2011) has been used before as a measure of overlapping. Given a dataset of instances in different classes, it quantifies the overlapping as a number between 0 and 1. To compute the Rvalue of a dataset, it considers each instance and its neighbourhood. An instance is said in *in overlapping* if too many of it nearest neighbours are labelled differently than him. The Rvalue of a dataset is then the proportion of instances *in overlapping*. The Rvalue thus has 2 parameters: the k-nearest neighbours to consider, and θ, the number of nearest neighbours from a different class above which an instance is *in overlapping*. Figure 2 shows the Rvalue for some UCI datasets, which shows overlapping occurs a lot in real-life datasets. For comparison, the cross dataset just above has an Rvalue of 0.41 for the same parameters.

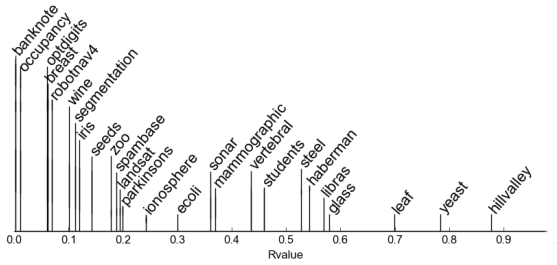


Figure 2. Rvalue of some UCI datasets, $k = 6$, $\theta = 1$.

To check our intuition that EM can handle overlapping better than SC, we look at the performance of these algorithm w.r.t. the Rvalue. Table 1 shows the average performance of these algorithm over some UCI datasets presented in further sections. Table 2 shows the same results but ignoring datasets were both algorithm performed badly. We assume that if both algorithm have an Adjusted Rand Index (Hubert & Arabie, 1985) (ARI) of less than 0.2, the dataset is not very suitable for clustering to begin with and we can then ignore it. A complete list of used datasets can be found in section 4.3. It can be seen that in that second case, EM performs better than SC when there is overlapping and vice versa when there is no or little overlapping. This difference is much reduced when including bad performing datasets. This suggest strongly that while overlapping does impact some algorithms more than others, other factors also have a significant influence of the performance of clustering algorithms.

	EM	SC
all	0.31	0.32
$Rvalue < 0.2$	0.48	**0.50**
$Rvalue > 0.2$	**0.19**	**0.19**

Table 1. Average clustering performance measured with ARI.

	EM	SC
all	0.45	0.47
$Rvalue < 0.2$	0.55	**0.59**
$Rvalue > 0.2$	**0.33**	0.31

Table 2. Same as table 1 for dataset where either EM or SC scored an ARI of at least 0.2.

4. Detecting overlap using constraints

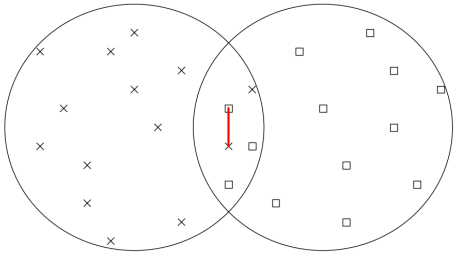
While the Rvalue is a good indicator of the extent to which clusters overlap, it is not useful in practice because it requires knowledge of the clusters, which we do not have. In this section, we present an alternative measure: the Constraint-Based Overlap value (CBO). The CBO is designed to correlate well with the Rvalue, while not requiring full knowledge of the clusters.

4.1. Definition

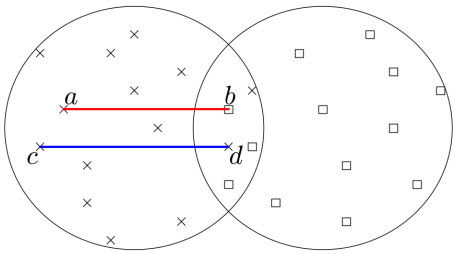
The CBO makes use of must-link and cannot-link constraints. The idea is to identify specific configurations of ML or CL constraints that indicate overlap. The CBO uses two configurations, illustrated in figure 3:

- short CL constraints: when two points are close together and yet belong to different clusters, this is an indication that the two clusters overlap in this area
- two parallel constraints, one of which is ML and the other CL, between points that are close. That is, if a and c are close to each other, and so are b and d, and a and b must link while c and d cannot link, then this implies overlapping, either around a and c or around b and d (see figure).

The more frequent those patterns, the more the clusters overlap. A limit case of the second configuration is when the 2 constraints involves the same point (e.g. $a = c$ on the figure) Then, by propagation of



(a) Short cannot-link pattern



(b) Parallel and close must-link and cannot-link pattern

Figure 3. Overlapping patterns in constraints. The crosses cluster and the squares cluster, both represented by a circle, overlap in the middle. A red line signifies a cannot-link, while a blue line signifies a must-link constraint.

the constraints, there is a short cannot-link constraint between the other 2 points.

The question is how to define "short" or "close". This has to be relative to "typical" distances. To achieve this, we introduce a kind of relative similarity measure, as follows. Let $d(x, x')$ be the distance between points x and x', and ϵ (ϵ') be the distance between x (x') and its k'th nearest neighbour. Then

$$s(x,x') = \begin{cases} 1 - \frac{d(x,x')}{max(\epsilon,\epsilon')} & if d(x,x') \leq max(\epsilon,\epsilon') \\ 0 & otherwise \end{cases}$$

That is: $s(x, x')$ is 1 when x and x' coincide, and linearly goes to 0, reaching 0 when $d(x, x') = \max(\epsilon, \epsilon')$, that is, x is no closer to x' than its k'th nearest neighbour, and vice versa.

Using this relative similarity, we can assign scores to both types of configurations mentioned above.

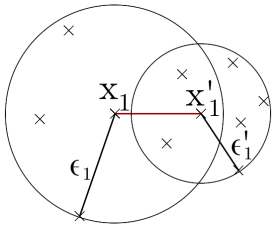
The score of a **short constraint** between two points x and x' is simply:

$$score(c) = s(x, x')$$

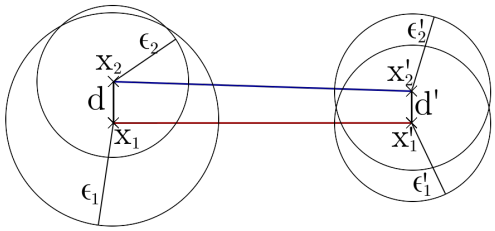
The score for a **pair of parallel constraints**, c_1 between points x_1 and x'_1 and c_2 between x_2 and x'_2, is computed as follows. Without loss of generality, assume $d(x_1, x_2) + d(x'_1, x'_2) \leq d(x_1, x'_2) + d(x'_1, x_2)$ (this can always be achieved by renaming x_2 to x'_2 and vice versa, see figure 4(b)). We then define:

$$score(c_1, c_2) = s(x_1, x_2) \times s(x'_1, x'_2)$$

The multiplication ensures that if either x_1 and x_2 or x'_1 and x'_2 are too far apart then the score is zero.



(a) Score of a single constraint



(b) Score of a pair of constraints

Figure 4. Scoring of single constraint(a) and pair of constraints(b) using the local similarity. The circles represent the neighbourhoods of the points.

In both cases, higher scores are more indicative of overlap. To have a measure for the whole dataset, we aggregate these scores over the whole constraint set. The idea is to compare the amount of short cannot-link constraints, direct (single pattern) or by propagation(double pattern), to the total amount of short constraints, both must-link and cannot-link. With CL the set of cannot-link constraints and ML the set of must-link constraints, we define

$$CBO = \frac{\sum_{c \in CL} score(c) + \sum_{\substack{c_1 \in CL \\ c_2 \in ML}} score(c_1, c_2)}{\sum_{c \in CL \cup ML} score(c) + \sum_{\substack{c_1 \in ML \\ c_2 \in CL \cup ML}} score(c_1, c_2)}$$

4.2. Stability

As one can imagine, the CBO can be very noisy for very small constraint sets. Several parameters influ-

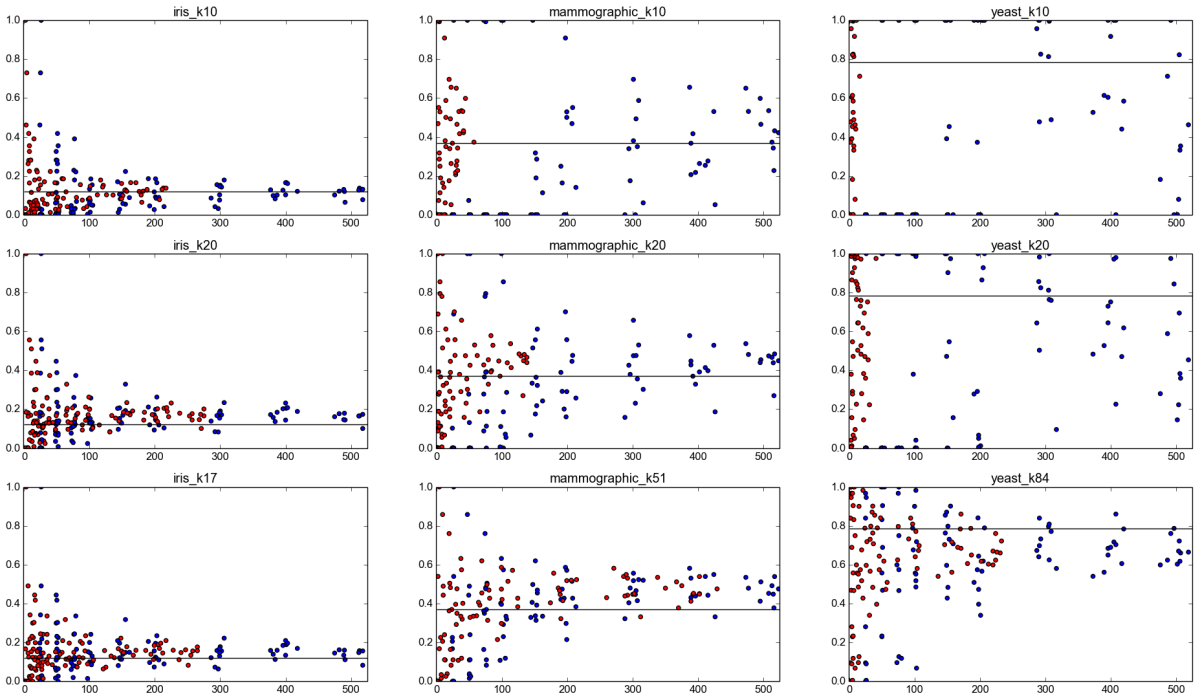


Figure 5. Convergence of the CBO w.r.t. the size of the constraint set. Three datasets are considered with increasing number of instances from left to right: iris(N=150), mammographic(N=830), yeast(N=1484). For each datasets, 80 constraint sets are sampled with various size (around 25,50,75,100,200,300,400,500). The CBO is computed for k=10 (top row), k=20 (middle row), k=10+N/20 (bottom row). The blue points correspond to the total number of constraints. The red points correspond to the number of constraints that actually participated in the measure. The Rvalue of the dataset (k=10, $\theta = 1$) is plotted as a black horizontal line.

ence that stability: the k-nearest neighbours to consider, the size of the dataset and the size of the constraint set. If the k is too small and the dataset too big, the measure would require too many constraints not to be noisy. To solve this problem, we need k to increase with the size of the dataset. For that reason, we set $k = 10 + N/20$ where N is the number of instances in the dataset. This has the desired effect while ensuring a minimal number of neighbours are considered for smaller datasets.

Figure 5 shows the variance of the CBO w.r.t. the size of the constraint set for 3 datasets of different sizes. For each dataset, several constraint sets of different sizes were sampled from the true labels. This shows that having a k increasing with the size of the dataset makes the CBO more stable.

4.3. Evaluation

The CBO is intended to serve as an alternative for the Rvalue, when the clusters are not known but some constraints are available. We therefore evaluate the CBO by comparing it to the Rvalue on a number of datasets from the UCI repository and the OpenML repository, namely *iris*, *glass*, *ionosphere*, *wine*, *vertebral*, *ecoli*, *seeds*, *students*, *robotnav4*, *yeast*, *zoo*, *breast cancer wisconsin*, *mammographic*, *banknote*, *haberman*, *segmentation*, *landsat*, *sonar*, *libras*, *hillvalley*, *optdigits*, *steel*, *leaf*, *spambase*, *parkinsons*, *occupancy*, *balance*, *pageblocks*, *diabetes*, *vehicle*, *authorship*, *ailerons*, *jedit*, *kc1*, *megawatt*, *blood*, *climate*, *fertility*, *heart*, *robotfail*, *volcanoes*, *engine*. For each dataset, 20 constraint sets of 200 random constraints were sampled. Figure 6 visualises how the Rvalue and the CBO (averaged over the 20 constraint sets) correlate, over the different datasets. This graph was produced for one particular value of k and θ for Rvalue, but other values give very similar results. With a correlation of 0.93, it is clear that CBO is useful as a proxy for Rvalue.

5. Algorithm selection

Now that we have the CBO to estimate overlap using constraints, we can use it for meta-learning, and more specifically algorithm selection. We picked 2 algorithms to select from: Expectation Maximization(EM) and Spectral Clustering(SC). We chose these two because among algorithms that build a global model like EM and algorithms that use local properties of the data like SC, these are 2 algorithms that perform the best on our datasets. To determine the performance of EM and SC, we ran the algorithms with different parameters and kept the best run. Then, we build 3 algorithm selection systems:

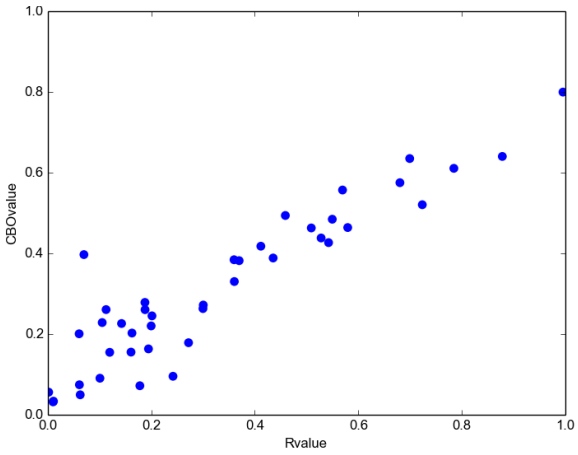


Figure 6. CBO with k=10+N/20 vs Rvalue with k=6 and θ=1.

- CBO: The first system only uses the CBO as meta-feature and choose EM if it is lower than 0.1, SC otherwise.
- Unsup: The second system uses unsupervised features that have been used by previous clustering algorithm selection system, and that are presented in table 3. As in (Ferrari & de Castro, 2015), we consider an attribute discrete if the number of distinct values observed for it is less than 30% of the number of instances. Using these meta-features, we learn a classifier to predict which of EM or SC will perform better.
- Full: The third system combines the unsupervised features and the CBO.

Unsupervised meta-feature description
Natural log of the number of instances
Natural log of the number of attributes
Percentage of outliers
Percentage of discrete attributes
Mean entropy of discrete attributes
Mean absolute correlation between distrete attributes
Mean skewness of continuous attributes
Mean kurtosis of continuous attributes
Mean absolute correlation between numerical attributes

Table 3. Unsupervised meta-features used for algorithm selection.

These 3 methods were run on the datasets presented in the previous section. For the constraint-based features, 20 constraint sets of about 200 constraints were sampled at random for each dataset. Table 5 shows the ARI averaged over datasets and constraint sets, using a leave-one-out cross validation for the 2 methods that

involved a classifier (Unsup and Ful). For those two methods, we used 3 classifiers: Support Vector Machine (SVM), Logistic Regression (LR) and Decision Trees (DT). For all algorithms (clustering, classifier, scores), we used the scikit-learn Python package (Pedregosa et al., 2011).

Classif.	EM	SC	CBO	Unsup	Full	Oracle
	0.31	0.32	0.33			0.37
SVM				0.33	0.33	
LR				0.33	0.33	
DT				0.33	0.31	

Table 4. Average ARI of multiple approaches: consistently EM or SC, selecting one of these using CBO, unsupervised features, or both ("Full"); and using an oracle to predict the best system.

Classif.	EM	SC	CBO	Unsup	Full	Oracle
	0.45	0.47	0.48			0.53
SVM				0.46	0.46	
LR				0.48	0.48	
DT				0.47	0.47	

Table 5. Same for datasets where either EM or SC scored an ARI of at least 0.2.

On average, algorithm selection methods perform a bit better than each algorithm separately. The improvement is quite modest, but relative to the maximum improvement possible (by using an oracle), still substantial. Interestingly, the CBO on its own performs as well as the whole set of features defined before. On the other hand, combining the CBO with those features does not further improve the results.

The choice of a threshold for the CBO method is rather flexible. We set it to 0.1 as it is a good value without being over-fitting. Figure 7 shows the variation of the performance of that method when varying that threshold for dataset with an ARI of at least 0.2 (which corresponds to the first line of table 5). It can be seen that any value between 0.1 and 0.3 has about the same score.

6. Conclusion

Algorithm selection and meta-learning have been studied mostly in the classification setting. In this paper, we have studied them in the context of clustering. Our main contributions are as follows.

First, we have identified *overlap between clusters* as a relevant property of the true clustering, meaning the clustering according to the true labels.

Second, because such overlap is difficult to quantify without knowing the cluster labels, we have proposed

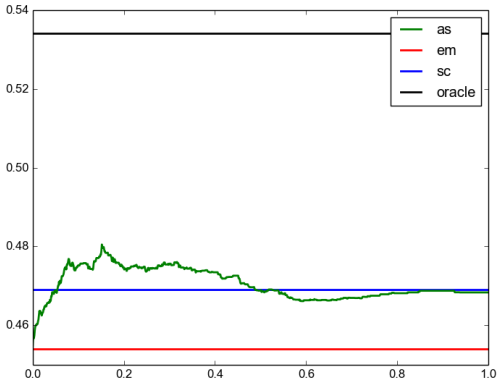


Figure 7. Performance of the CBO algorithm selection (AS) when the threshold for choosing EM or SC varies from 0 to 1 on the x axis.

a measure called CBO, which uses information from must-link / cannot-link constraints to estimate the amount of overlap. We have shown that the CBO correlates well with the Rvalue, a previously proposed measure for overlap in a completely known clustering. As such, the CBO can be a useful measure in itself, also outside the context of algorithm selection for clustering.

Third, we have empirically estimated the usefulness of selecting the most appropriate clustering method, among two methods with quite different properties: EM, which is good at detecting overlapping clusters but finds only elliptic clusters, and SC, which can find clusters of any shape but cannot return overlapping clusters. The conclusion is that the CBO is indeed informative for selecting the best among these two; it yields a small but noticeable improvement, and this improvement is comparable to the improvement obtained by using a set of 10 unsupervised features previously proposed for clustering algorithm selection. When combined with those other features, however, the CBO does not yield a further improvement. This suggests that the information contained in the CBO is already contained in the other features.

Compared to choosing the best clustering method using an oracle, CBO-based selection leaves room for further improvement. This is perhaps not surprising, given that the amount of overlap among clusters is one aspect that determines the effectiveness of clustering methods, but certainly not the only one. An indication of cluster shapes, for instance, is likely to give additional information. The question remains open to which extent this and other features can be derived

from constraints, and to what extent this can lead to better clustering algorithm selection.

Acknowledgments

Research financed by the KU Leuven Research Council through project IDO/10/012.

References

Adam, A., & Blockeel, H. (2015). Dealing with overlapping clustering: a constraint-based approach to algorithm selection. *Meta-learning and Algorithm Selection workshop-ECMLPKDD2015* (pp. 43–54).

Babaki, B., Guns, T., & Nijssen, S. (2014). Constrained clustering using column generation. In *Integration of ai and or techniques in constraint programming*, 438–454. Springer.

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2005). Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, *6*, 937–965.

Bilenko, M., Basu, S., & Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. *Proceedings of the twenty-first international conference on Machine learning* (p. 11).

De Souto, M. C., Prudencio, R. B., Soares, R. G., De Araujo, R. G., Costa, I. G., Ludermir, T. B., Schliep, A., et al. (2008). Ranking and selecting clustering algorithms using a meta-learning approach. *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on* (pp. 3729–3735).

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1–38.

Duong, K.-C., Vrain, C., et al. (2015). Constrained clustering by constraint programming. *Artificial Intelligence*.

Ferrari, D. G., & de Castro, L. N. (2015). Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. *Information Sciences*, *301*, 181–194.

Hu, P., Vens, C., Verstrynge, B., & Blockeel, H. (2013). Generalizing from example clusters. *Discovery Science* (pp. 64–78).

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of classification*, *2*, 193–218.

Oh, S. (2011). A new dataset evaluation method based on category overlap. *Computers in Biology and Medicine*, *41*, 115–122.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pelleg, D., & Baras, D. (2007). K-means with large and noisy constraint sets. In *Machine learning: Ecml 2007*, 674–682. Springer.

Prudêncio, R. B., De Souto, M. C., & Ludermir, T. B. (2011). Selecting machine learning algorithms using the ranking meta-learning approach. In *Meta-learning in computational intelligence*, 225–243. Springer.

Ruiz, C., Spiliopoulou, M., & Menasalvas, E. (2007). C-dbscan: Density-based clustering with constraints. In *Rough sets, fuzzy sets, data mining and granular computing*, 216–223. Springer.

Soares, R. G., Ludermir, T. B., & De Carvalho, F. A. (2009). An analysis of meta-learning techniques for ranking clustering algorithms applied to artificial data. In *Artificial neural networks–icann 2009*, 131–140. Springer.

Van Craenendonck, T., & Blockeel, H. (2016). Constraint-based clustering selection. *arXiv preprint arXiv:1609.07272*.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, *17*, 395–416.

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al. (2001). Constrained k-means clustering with background knowledge. *ICML* (pp. 577–584).

Wang, X., & Davidson, I. (2010). Flexible constrained spectral clustering. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 563–572).

Conference Track

Extended Abstracts

A Probabilistic Modeling Approach to Hearing Loss Compensation

Thijs van de Laar[1] T.W.V.D.LAAR@TUE.NL
Bert de Vries[1,2] BDEVRIES@IEEE.ORG
[1]Department of Electrical Engineering, Eindhoven University of Technology
[2]GN Hearing, Eindhoven

Keywords: Hearing Aids, Hearing Loss Compensation, Probabilistic Modeling, Bayesian Inference

1. Introduction

Hearing loss is a serious and prevalent condition that is characterized by a frequency-dependent loss of sensitivity for acoustic stimuli. As a result, a tone that is audible for a normal-hearing person might not be audible for a hearing-impaired patient. The goal of a hearing aid device is to restore audibility by amplification and compressing the dynamic range of acoustic inputs to the remaining audible range of the patient. In practice, current hearing aids apply frequency- and intensity-dependent gains that aim to restore normal audibility levels for the impaired listener.

The hearing aid algorithm design problem is a difficult engineering issue with many trade-offs. Each patient has her own auditory loss profile and individual preferences for processed audio signals. Yet, we cannot afford to spend intensive tuning sessions with each patient. As a result, there is a need for automating algorithm design iterations based on in-situ collected patient feedback.

This short paper summarizes ongoing work on a probabilistic modeling approach to the design of personalized hearing aid algorithms (van de Laar & de Vries, 2016). In this framework, we first specify a probabilistic generative model that includes an explicit description of the hearing loss problem. Given the model, hearing aid signal processing relates to on-line Bayesian state estimation (similar to Kalman filtering). Estimation of the tuning parameters (known as the 'fitting' task in hearing aid parlance) corresponds to Bayesian parameter estimation. The innovative aspect of the framework is that both the signal processing and fitting tasks can be *automatically* inferred from the probabilistic model in conjunction with patient appraisals (the data). The architecture of our design loop is shown in Fig. 1.

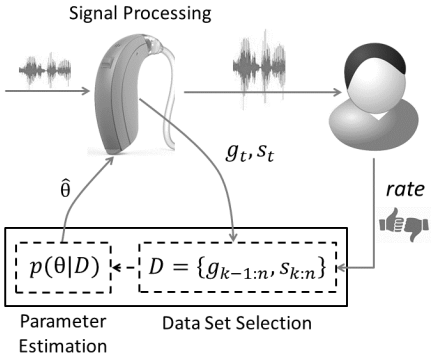


Figure 1. The iterative algorithm design loop, featuring the interplay between signal processing (Eq.5) and parameter estimation (Eq.6). Tuning parameters are designated by θ. Figure adapted from (van de Laar & de Vries, 2016).

2. Model Specification

We describe the hearing loss compensation model for one frequency band. In practice, a hearing aid would apply the derived algorithms to each band independently. For a given patient wearing hearing aids, we define the received sound level as

$$r_t = \mathrm{L}(s_t + g_t; \phi) \tag{1}$$

where s_t is the sound pressure level (in dB SPL) of the input signal that enters the hearing aid, g_t is the hearing aid gain and L is a function with tuning parameters ϕ that models the patient's hearing impairment in accordance with (Zurek & Desloge, 2007).

Hearing loss compensation balances two simultaneous constraints. First, we want restored sound levels to be approximately experienced at normal hearing levels:

$$s_t|g_t \sim \mathcal{N}(r_t, \vartheta) = \mathcal{N}(\mathrm{L}(s_t + g_t; \phi), \vartheta)\ . \tag{2}$$

Appearing in *Proceedings of Benelearn 2017*.

Secondly, in order to minimize acoustic signal distortion, the compensation gain should remain as constant as possible, which we model as

$$g_t|g_{t-1} \sim \mathcal{N}(g_{t-1}, \varsigma) \ . \tag{3}$$

The trade-off between conditions Eqs. 2 and 3 is controlled by the noise variances ϑ and ς. The full generative model is specified by combining Eqs. 2 and 3:

$$p(g_{0:T}, s_{1:T}, \varsigma, \vartheta, \phi) = \tag{4}$$
$$p(g_0)\, p(\varsigma)\, p(\vartheta)\, p(\phi) \prod_{t=1}^{T} p(s_t|g_t, \phi, \vartheta)\, p(g_t|g_{t-1}, \varsigma) \, .$$

In this model, s_t is an observed input sequence, g_t is the hidden gain signal, and $\theta = \{\varsigma, \vartheta, \phi\}$ are tuning parameters.

3. Signal Processing and Fitting as Probabilistic Inference

The signal processing and parameter estimation algorithms follow by applying Bayesian inference to the generative model. The hearing aid signal processing algorithm is defined by estimating the current gain g_t from given past observations $s_{1:t}$ and given parameter settings $\theta = \hat{\theta}$. In a Bayesian framework, this amounts to computing

$$p(g_t|s_{1:t}, \hat{\theta}) = \frac{\int \cdots \int p(g_{0:t}, s_{1:t}, \hat{\theta})\, \mathrm{d}g_0 \ldots \mathrm{d}g_{t-1}}{\int \cdots \int p(g_{0:t}, s_{1:t}, \hat{\theta})\, \mathrm{d}g_0 \ldots \mathrm{d}g_t} \, . \tag{5}$$

A suitable personalized parameter setting is vital to satisfactory signal processing. Bayesian parameter estimation amounts to computing

$$p(\theta|D) = \frac{p(g_{k-1:n}, s_{k:n}, \theta)}{\int p(g_{k-1:n}, s_{k:n}, \theta)\, \mathrm{d}\theta} \, . \tag{6}$$

In this formula, we assume availability of a training set of pairs $D = \{(g_{k-1:n}, s_{k:n})\}$, where k and $n > k$ are positive indices. This training set can be obtained from in-situ collected patient appraisals on the quality of the currently selected hearing aid algorithm (Fig.1). After the user casts a positive appraisal, we collect a few seconds of both the hearing aid input signal and corresponding gain signals and add these signal pairs to the training database.

4. Inference Execution through Message Passing

Equations (5) and (6) are very difficult to compute directly. We have developed a software toolbox to automate these inference problems by message passing in a Forney-style Factor Graph (FFG) (Forney, 2001). In an FFG, nodes correspond to factors and edges represent variables. The FFG for the generative model of Eq. 4 is depicted in Fig. 2. The arrows indicate the message passing schedule that recursively executes the signal processing inference problem of Eq. 5. Particular message passing update rules were derived in accordance with (Loeliger, 2007) and (Dauwels, 2007).

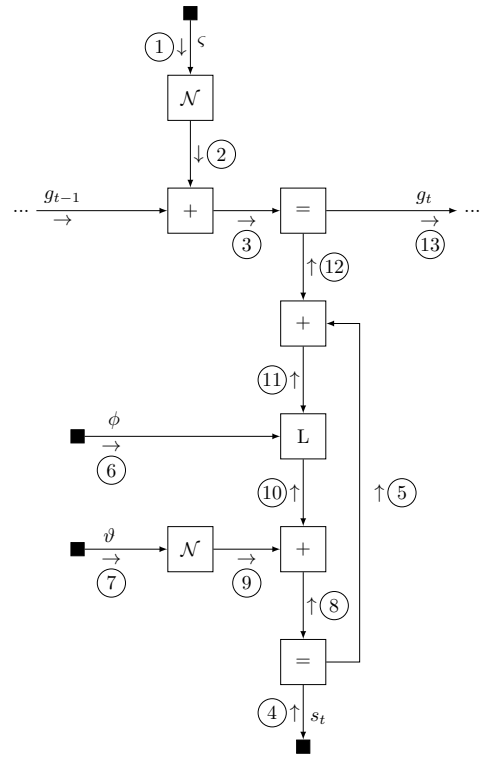


Figure 2. A Forney-style factor graph for one time step in the generative model. The small numbered arrows indicate a recursive message passing schedule for executing the signal processing task of Eq. 5. Figure adapted from (van de Laar & de Vries, 2016).

Simulations show that the inferred signal processing algorithm exhibits compressive amplification behavior that is similar to the manually designed dynamic range compression circuits in hearing aids. Simulations also verify that the parameter estimation algorithm is able to recover preferred tuning parameters from a user-selected training example.

Crucially, our algorithms for signal processing and fitting can be automatically inferred from a given model plus in-situ collected patient appraisals. Therefore, in contrast to existing design methods, this approach allows for hearing aid personalization by a patient without need for human design experts in the loop.

References

Dauwels, J. (2007). On Variational Message Passing on Factor Graphs. *IEEE International Symposium on Information Theory* (pp. 2546–2550).

Forney, G.D., J. (2001). Codes on graphs: normal realizations. *IEEE Transactions on Information Theory*, *47*, 520–548.

Loeliger, H.-A. (2007). Factor Graphs and Message Passing Algorithms – Part 1: Introduction.

van de Laar, T., & de Vries, B. (2016). A probabilistic modeling approach to hearing loss compensation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*, 2200–2213.

Zurek, P. M., & Desloge, J. G. (2007). Hearing loss and prosthesis simulation in audiology. *The Hearing Journal*, *60*, 32–33.

An In-situ Trainable Gesture Classifier

Anouk van Diepen[1] A.V.DIEPEN@STUDENT.TUE.NL
Marco Cox[1] M.G.H.COX@TUE.NL
Bert de Vries[1,2] BDEVRIES@GNRESOUND.COM
[1]Department of Electrical Engineering, Eindhoven University of Technology, [2]GN Hearing BV, Eindhoven

Keywords: gesture recognition, probabilistic modeling, Bayesian inference, empirical Bayes.

1. Introduction

Gesture recognition, i.e., the recognition of pre-defined gestures by arm or hand movements, enables a natural extension of the way we currently interact with devices (Horsley, 2016). Commercially available gesture recognition systems are usually pre-trained: the developers specify a set of gestures, and the user is provided with an algorithm that can recognize just these gestures.

To improve the user experience, it is often desirable to allow users to define their own gestures. In that case, the user needs to train the recognition system herself by a set of example gestures. Crucially, this scenario requires learning gestures from just a few training examples in order to avoid overburdening the user.

We present a new in-situ trainable gesture classifier based on a hierarchical probabilistic modeling approach. Casting both learning and recognition as probabilistic inference tasks yields a principled way to design and evaluate algorithm candidates. Moreover, the Bayesian approach facilitates learning of prior knowledge about gestures, which leads to fewer needed examples for training new gestures.

2. Probabilistic modeling approach

Under the probabilistic modeling approach, both learning and recognition are problems of probabilistic inference in the same generative model. This generative model is a joint probability distribution that specifies the relations among all (hidden and observed) variables in the model.

Let $y = (y_1, ..., y_T)$ be a time series of measurements corresponding to a single gesture with underlying characteristics θ. The characteristics are unique to gestures of type (class) k. We can capture these dependencies by the probability distribution

$$p(y, \theta, k) = \underbrace{p(y|\theta)}_{\substack{\text{dynamical}\\ \text{model}}} \cdot \underbrace{p(\theta|k)}_{\substack{\text{gesture}\\ \text{characteristics}}} \cdot \underbrace{p(k)}_{\substack{\text{gesture}\\ \text{class index}}} . \quad (1)$$

Because the measurement sequence is temporally correlated, it is natural to choose $p(y|\theta)$ to be a hidden Markov model (HMM). HMMs have been successfully applied to gesture classification in the past (Mäntylä et al., 2000). Under this model, θ represents the set of parameters of the HMM.

During learning, the parameter values θ of gestures of class k need to be learned from data. We choose to learn this distribution using a two-step approach.

In the first step, a prior for θ is constructed. This prior distribution can be obtained in various ways. We have chosen to construct one that captures the common characteristics that are shared among *all* gestures. This is done by learning the distribution using dataset $\mathcal{D}$, consisting of one measurement from each gesture class. This can be expressed as

$$p(\theta|\mathcal{D}, k) = \frac{p(\mathcal{D}, \theta, k)}{\int p(\mathcal{D}, \theta, k)\, \mathrm{d}\theta} . \quad (2)$$

In the second step, the parameter distribution is learned for a specific gesture class, using the previously learned $p(\theta|\mathcal{D}, k)$ and a set of measurements $\mathcal{D}_k$ with the same class k:

$$p(\theta|\mathcal{D}, \mathcal{D}_k, k) = \frac{p(\mathcal{D}_k|\theta)p(\theta|\mathcal{D}, k)p(k)}{\int p(\mathcal{D}_k, \theta, k|\mathcal{D})\, \mathrm{d}\theta} . \quad (3)$$

In practice, exact evaluation of Eq. 2 and Eq. 3 is intractable for our model due to the integral in the denominator. We use variational Bayesian inference to approximate this distribution (MacKay, 1997), which

results in a set of update equations that need to be iterated until convergence.

During recognition, the task of the algorithm is to identify the gesture class with the highest probability of having generated the measurement y. This is expressed by

$$p(k|y) = \frac{\int p(y, \theta, k)\, \mathrm{d}\theta}{\sum_k \int p(y, \theta, k)\, \mathrm{d}\theta}\,. \tag{4}$$

If we assume that each gesture is performed with the same *a priori* probability $p(k)$, then $p(y|k) \propto p(k|y)$. To calculate $p(y|k)$, the method as proposed in Chapter 3 of Beal (2003) is used: the obtained variational posterior distribution of the parameters is replaced by its mean, which allows exact evaluation of $p(y|k)$.

3. Experimental validation

We built a gesture database using a Myo sensor bracelet (ThalmicLabs, 2016), which is worn just below the elbow (see Fig. 1). The Myo's inertial measurement unit measures the orientation of the bracelet. This orientation signal is sampled at 6.7 Hz, converted into the direction of the arm, and quantized using 6 quantization directions. The database contains 17 different gesture classes, each performed 20 times by the same user. The duration of the measurements was fixed to 3 seconds.

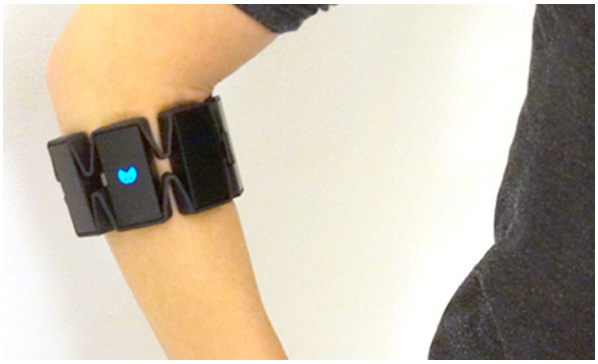


Figure 1. The Myo sensor bracelet used to measure gestures.

As a measure of performance, we use the recognition rate defined as:

$$\text{Recognition rate} = \frac{\#\text{ correctly classified}}{\text{total }\#\text{ of samples}}\,. \tag{5}$$

The gesture database is split in a training set containing 5 samples of every gesture class, and a test set containing the remaining (15x17=) 255 samples. The recognition rate is evaluated on models trained on 1 through 5 examples. To minimize the influence of the training order, the results are averaged over 5 different permutations of the training set.

To compare our algorithm, we have also evaluated the recognition rate of the same algorithm with uninformative prior distributions and of a 1-Nearest Neighbor (1-NN) algorithm using the same protocol.

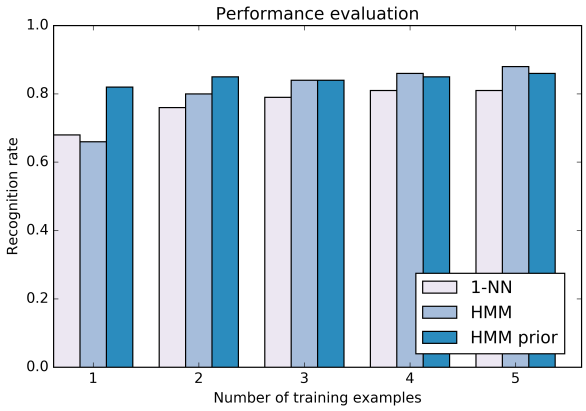


Figure 2. Recognition rates of the 1-NN algorithm, the proposed algorithm without prior information (HMM), and the proposed algorithm with informed prior distributions (HMM prior).

Figure 2 shows the recognition rates of the algorithms. Both hidden Markov based algorithms have a higher recognition rate than the 1-NN algorithm. For personalization of gesture recognition, we are especially interested in learning gesture classes using a low number of training examples. In particular for one-shot training (from one example only), the hidden Markov model using the learned prior distribution corresponds to the highest recognition rate.

The algorithm was also tested for gestures that are not used to learn the prior distribution. When the prior is constructed with similar gestures, the new gestures are also learned faster than when uninformative priors are used.

There are multiple ways to incorporate these results in a practical gesture recognition system. For example, the prior distribution can be constructed by the developers of the algorithm. Another possibility is to allow users to provide prior distributions themselves. This means that the system will take longer to set up, but when a user wants to learn a specific gesture under in-situ conditions, it will require less training examples.

References

Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. Doctoral dissertation, University College London.

Horsley, D. (2016). Wave hello to the next interface. *IEEE Spectrum*, *53*, 46–51.

MacKay, D. J. C. (1997). *Ensemble Learning for Hidden Markov Models* (Technical Report).

Mäntylä, V.-M., Mäntyjärvi, J., Seppänen, T., & Tuulari, E. (2000). Hand gesture recognition of a mobile device user. *2000 IEEE International Conference on Multimedia and Expo.* (pp. 281–284).

ThalmicLabs (2016). Myo Gesture Control Armband. Retrieved from https://www.myo.com/.

Text mining to detect indications of fraud in annual reports worldwide

Marcia Fissette FISSETTE.MARCIA@KPMG.NL
KPMG, Laan van Langerhuize 1, 1186 DS , Amstelveen, The Netherlands

Bernard Veldkamp B.P.VELDKAMP@UTWENTE.NL
University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands

Theo de Vries TDEVRIES@WXS.NL
University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands

Keywords: Text mining, Fraud, Annual reports

1. Introduction

Fraud affects the financial results presented in the annual reports of companies worldwide. Analysis performed on annual reports focuses on the quantitative data in these reports. However, the amount of textual information in annual reports increased in the past decade with companies using the reports to project themselves. The texts provide information that is complementary to the financial results. Therefore, the analysis of the textual information in annual reports may provide indications of the presence of fraud within a company. This piece of research uses the extensive and reality approaching data set containing annual reports of companies worldwide to answer the research question:

Can a text mining model be developed that can detect indications of fraud in the management discussion and analysis section of annual reports of companies worldwide?

2. The data

We selected the fraud cases in the period from 1999 to 2013 from news messages and the Accounting and Auditing Enforcement Releases (AAER's) published by the Securities and Exchange Commission (SEC). The selection process results in 402 annual reports in the fraud set. For each annual report in the fraud set we collect annual reports of companies similar to the companies in the fraud set, but for which no fraudulent activities are known. The latter category is referred to as the no-fraud reports. We match the fraud and no-fraud reports on year, sector and number of employees. The matching process results in 1.325 annual reports which do not contain known fraud. The resulting data set contains annual reports in the period from 1999 to 2011. The formats of these annual reports differs depending on the on the stock exchange to which the organization is listed or the country of origin. Filings to the SEC are on forms 10-K or 20-F while others have a freer format.

It is argued that the Management Discussion and Analysis (MD&A) section is the most read part of the annual report (Li, 2010). Previous research on 10-K reports showed promising results (Cecchini et al., 2010; Glancy & Yadav, 2011; Purda & Skillicorn, 2010; Purda & Skillicorn, 2012; Purda & Skillicorn, 2015) Therefore, the MD&A section is a good starting point to determine whether text mining is a suitable means for detecting indications of fraud in annual reports worldwide. The manual extraction of the MD&A sections is a labor-intensive task. Therefore, we developed an algorithm that is able to detect the start of the MD&A section based on the section headers. The MD&A section is not always explicitly present in the free format annual reports. From these reports we selected the sections that most closely correspond to the MD&A section.

3. The text mining model

Before extracting the features for the text mining model, graphs, figures and tables are excluded because their primary way to convey information is not based on text. We use the tokenizers from the Natural Lan-

The original paper is under review.

guage Toolkit (NLTK) for Python to identify sentences and words (Bird & Klein, 2009). HTML-tags in forms 10-K and 20-F are removed using the Python package 'BeautifulSoup'

We develop a baseline model comprising of word unigrams. To obtain an informative set of word unigrams we exclude stop words and stem the words using the Porter stemmer in NLTK (Bird & Klein, 2009). Words that appear only in one MD&A section in the entire data set are not informative. Therefore these words will not be used as features. Furthermore, we apply 'term frequency-inverse document frequency' (TF-IDF) as a normalization step of the word counts to take into account the length of the text and the commonality of the word in the entire data set (Manning & Schütze, 1999). Finally, the chi squared method is applied to select the most informative features. We start with the top 1.000 and increase the number of features in steps of 1.000 until 24.000 to find the optimal number of features.

The Naïve Bayes classifier (NB) and Support Vector Machine (SVM) have been proven successful in text classification tasks in several domains (Cecchini et al., 2010; Conway et al., 2009; Glancy & Yadav, 2011; Goel et al., 2010; He & Veldkamp, 2012; Joachims, 1998; Manning & Schütze, 1999; Metsis et al., 2006; Purda & Skillicorn, 2015). Therefore, this research uses these two types of machine learning approaches to develop a baseline text mining model. Using 10-fold stratified cross validation on 70% of the data the data is split into train and test sets. The remaining 30% is saved for the best performing model in de development phase.

A word unigrams approach is a limited way of looking at texts because it omits a part of the textual information, such as the grammar. Therefore, we extend the baseline model with linguistic features categories to determine whether other types of textual information may improve the results of the baseline model. The first category consists of descriptive features, this includes the number of words and the number of sentences in the text. The second category of features represents the complexity of a text. Examples of these features are the average sentence length and the percentage of long words. The third group of captures the grammatical information, such as the percentage of verbs, nouns and several types of personal pronouns. The fourth category assesses the readability of the text by using readability scores, including the 'Flesch Reading Ease Score'. The fifth categories measures psychological processes such as positive and negative sentiment words. Finally, we include words bigrams and grammatical relations between two words, extracted with the Stanford parser, as features to the model (De Marneffe et al., 2006).

4. Results

Figure 1 shows the accuracy of the NB and SVM baseline models. For both types of models the optimal number of features is around 10.000 unigrams. With an accuracy of 89% the NB model outperforms the SVM that achieves an accuracy of 85%.

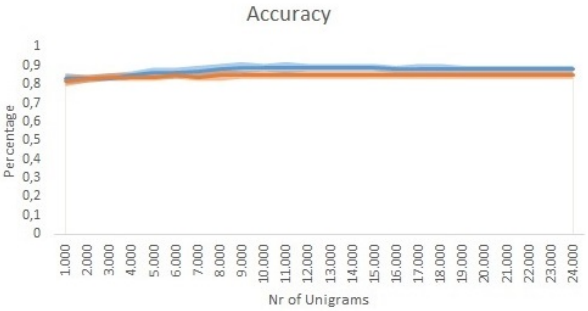


Figure 1. Performance of the Naïve Bayes and Support Vector Machine models.

The linguistic features of the descriptive, complexity, grammatical, readability and psychological process categories did not improve the result of the baseline models. The performance on the test set of the SVM model increased to 90% by adding the most informative bigrams. The addition of the relation features did not further increase the performance.

5. Discussion and conclusion

The results show that it is possible to use text mining techniques to detect indications of fraud in the management discussion and analysis section of annual reports of companies worldwide. The word unigrams capture the majority of the subtle information that differentiates fraudulent from non fraudulent annual reports. The additional information that the linguistic information provides is very limited, an only attributable to the bigrams. Additional research may address the effects of the random 10-fold splitting process, the effects of multiple authors on linguistic features of a text and the possibilities of a ensemble of machine learning algorithms for detecting fraud in annual reports worldwide.

References

Bird, Steven, E. L., & Klein, E. (2009). *Natural language processing with python*. OReilly Media Inc.

Cecchini, M., Aytug, H., Koehler, G. J., & Pathak, P. (2010). Making words work: Using financial text as a predictor of financial events. *Decision Support Systems*, *50*, 164 – 175.

Conway, M., Doan, S., Kawazoe, A., & Collier, N. (2009). Classifying disease outbreak reports using n-grams and semantic features. *International journal of medical informatics*, *78*, e47–e58.

De Marneffe, M.-C., MacCartney, B., Manning, C. D., et al. (2006). Generating typed dependency parses from phrase structure parses. *Proceedings of LREC* (pp. 449–454).

Glancy, F. H., & Yadav, S. B. (2011). A computational model for financial reporting fraud detection. *Decision Support Systems*, *50*, 595–601.

Goel, S., Gangolly, J., Faerman, S. R., & Uzuner, O. (2010). Can linguistic predictors detect fraudulent financial filings? *Journal of Emerging Technologies in Accounting*, *7*, 25–46.

He, Q., & Veldkamp, D. B. (2012). Classifying unstructured textual data using the product score model: an alternative text mining algorithm. In T. Eggen and B. Veldkamp (Eds.), *Psychometrics in practice at rcec*, 47 – 62. Enschede: RCEC.

Joachims, T. (1998). *Text categorization with support vector machines: Learning with many relevant features*. Springer.

Li, F. (2010). The information content of forward-looking statements in corporate filingsa naïve bayesian machine learning approach. *Journal of Accounting Research*, *48*, 1049–1102.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA, USA: MIT Press.

Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naive bayes-which naive bayes? *CEAS* (pp. 27–28).

Purda, L., & Skillicorn, D. (2012). Accounting variables, deception, and a bag of words: Assessing the tools of fraud detection. *Available at SSRN: http://ssrn.com/abstract=1670832*.

Purda, L., & Skillicorn, D. (2015). Accounting variables, deception, and a bag of words: Assessing the tools of fraud detection. *Contemporary Accounting Research*, *32*, 1193–1223.

Purda, L. D., & Skillicorn, D. (2010). Reading between the lines: Detecting fraud from the language of financial reports. *Available at SSRN: http://ssrn.com/abstract=1670832*.

Do you trust your multiple instance learning classifier?

Veronika Cheplygina[1,2] V.CHEPLYGINA@TUE.NL
Lauge Sørensen[3]
David M. J. Tax[4]
Marleen de Bruijne[2,3]
Marco Loog[4,3]
[1] Medical Image Analysis group, Eindhoven University of Technology, The Netherlands
[2] Biomedical Imaging Group Rotterdam, Erasmus Medical Center, Rotterdam, The Netherlands
[3] The Image Section, University of Copenhagen, Copenhagen, Denmark
[4] Pattern Recognition Laboratory, Delft University of Technology, The Netherlands

Keywords: multiple instance learning

Abstract

Multiple instance learning (MIL) is a weakly supervised learning scenario where labels are given only for groups (bags) of examples (instances). Because some MIL classifiers can provide instance labels at test time, MIL is popular in applications where labels are difficult to acquire. However, MIL classifiers are frequently only evaluated on their bag-level, not instance-level performance. In this extended abstract, which covers previously published work, we demonstrate why this could be problematic and discuss this open problem.

1. Introduction

Consider the task of training a classifier to segment a medical image into abnormal and healthy image patches. Traditionally, we would need examples of both healthy and abnormal patches to train a supervised classifier. In medical imaging, obtaining such ground truth patch labels is often difficult, but image labels, such as the diagnosis of the patient, are more easily available. **The lack of ground truth patch labels calls for weakly-supervised approaches**, such as multiple instance learning (MIL).

In MIL, we are only given bags which are labeled positive or negative. Positive bags are often assumed to have a few positive instances and negative bags are assumed to have only negative instances. In our example, positive bags are images with an abnormal diagnosis (and hence abnormal patches), while negative bags are images of healthy subjects. We can then train a classifier to distinguish positive from negative bags as well as possible. In some cases while the classifier is learning to classify bags, it also learns to classify instances, providing the sought-after patch labels. It's like music to our ears - we only need (easy) image labels as input, but we get (difficult) patch labels as output. It's therefore not surprising that MIL is gaining popularity in medical image analysis - see (Quellec et al., 2017) for a recent survey.

Now that we trained a MIL classifier with image labels, we would like to evaluate it. Since we trained our classifier to classify bags, we evaluate it on its ability to classify bags in the test set. We then choose the best bag classifier as the classifier that will give us those elusive patch labels. Where this goes wrong, is that the best bag classifier is typically not the best instance classifier (Vanwinckelen et al., 2016). Consider a test image with patches {A, B, C}, of which only A is abnormal. Classifiers which classify any subset of { {A}, {B}, {C}, {A,B}, {A,C}, {B,C}, {A,B,C} } as abnormal are equally good classifiers for our test image, but have very different performances on patch-level! Therefore, **evaluating a weakly-supervised approach calls for ground-truth patch labels**.

2. Methods and Results

If we are lucky, we can ask an expert to label a part of the patches in the test image, or perhaps, just to

Preliminary work. Under review for Benelearn 2017. Do not distribute.

visually assess the results. But what can we do if this isn't possible? The approach we proposed in our previous work (Cheplygina et al., 2015), is to invent unsupervised patch-level evaluation measures which do not need any patch labels. We reasoned that, if a classifier is finding the true patch labels, it should find similar patch labels, even if we change the classifier slightly. If the classifier is finding different patch labels every time, we probably don't want to trust it. By changing the classifier slightly and evaluating the *stability* of the patch labels, we get a different sense of how well the classifier is doing.

Let z_i and z'_i be the patch-level outputs of the same type of classifier, that has been slightly changed. Then we can define a very simple stability measure as

$$S_+(\mathbf{z}, \mathbf{z}') = n_{11}/(n_{01} + n_{10} + n_{11}) \qquad (1)$$

where $n_{00} = |\{i|z_i = 0 \wedge z'_i = 0\}|$, $n_{01} = |\{i|z_i = 0 \wedge z'_i = 1\}|$, $n_{10} = |\{i|z_i = 1 \wedge z'_i = 0\}|$ and $n_{11} = |\{i|z_i = 1 \wedge z'_i = 1\}|$. In other words, we measure the agreement of the classifiers on patches, that either of the two considered positive.

In our experiments we used six MIL datasets: Musk1, Musk2, Breast, Messidor, COPD validation and COPD test. Musk1 and Musk2 are MIL benchmark datasets, while the others are medical imaging datasets. We split each dataset into a training set and a test set.

We trained eight types of MIL classifiers: simpleNM, miNM, simple1NN, mi1NN, simpleSVM, miSVM, MILES and MILBoost. To change each classifier slightly, we trained it on 10 random samples of 80% of the training bags. We evaluated these 10 classifier versions on the fixed test set, and computed two measures: the bag-level performance, and the instance-level stability, averaged over all $\frac{1}{2}10(10-1) = 45$ pairs of the 10 slightly changed versions of the same classifier.

In Fig. 1 we plot the bag performance against the instance-level stability. We see that the classifier with the best bag performance is not always the most stable classifier. For example, for Musk1 dataset, the best bag classifier is MILES. But, taking into account the instance labels, we might want to choose mi1NN, sacrificing a little bit of bag performance, but gaining a lot of stability.

3. Conclusions

The take-home message is that if we use MIL to classify instances, we should be careful about how we evaluate the classifier - only bag performance might not be sufficient, and the instance labels given by the best bag classifier might not be reliable. A possible solution is to look at additional, unsupervised evaluation measures, such as instance label stability.

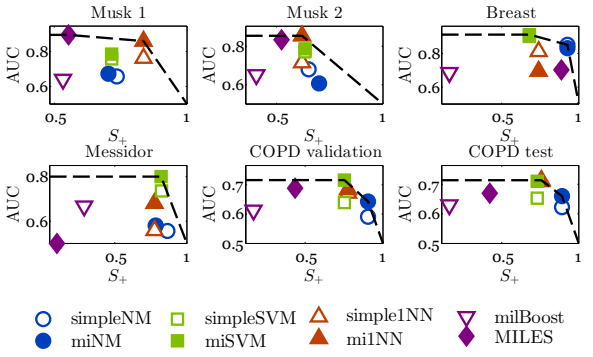


Figure 1. Bag performance (AUC) vs instance stability for six datasets (each plot) and eight types of MIL classifiers (each marker). The dotted lines show the Pareto frontiers, which indicate the "best" classifiers if both bag AUC and instance stability are considered. Figure from (Cheplygina et al., 2015), with permission.

However, there is still room for improvement. While low stability makes us doubt the classifier, high stability doesn't inspire confidence - for example, a classifier that always outputs 0 is very stable using our measure. In future work, we want to find a stability measure that is informative about the instance performance. But to validate such a measure, we run into the same problem: getting enough data with ground truth instance labels.

References

Cheplygina, V., Sørensen, L., Tax, D. M. J., de Bruijne, M., & Loog, M. (2015). Label stability in multiple instance learning. *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 539–546).

Quellec, G., Cazuguel, G., Cochener, B., & Lamard, M. (2017). Multiple-instance learning for medical image and video analysis. *IEEE Reviews in Biomedical Engineering*, in press.

Vanwinckelen, G., Fierens, D., Blockeel, H., et al. (2016). Instance-level accuracy versus bag-level accuracy in multi-instance learning. *Data Mining and Knowledge Discovery, 30*, 313–341.

A Gaussian process mixture prior for hearing loss modeling

Marco Cox[1] M.G.H.COX@TUE.NL
Bert de Vries[1,2] BDEVRIES@IEEE.ORG
[1]Dept. of Electrical Engineering, Eindhoven University of Technology, [2]GN Hearing, Eindhoven.

Keywords: hearing loss, probabilistic modeling, Bayesian machine learning.

1. Introduction

The most common way to quantify hearing loss is by means of the *hearing threshold*. This threshold corresponds to the lowest sound intensity that the person in question can still perceive, and it is a function of frequency. The typical process of measuring the hearing threshold is known as *pure-tone audiometry* (Yost, 1994), and it usually consists of incrementally estimating the threshold value at a set of standard frequencies ranging from 125 Hz to 8 kHz using a staircase "up 5 dB – down 10 dB" approach.

A recent line of work in the field of machine learning has focused on improving the efficiency of hearing loss estimation by taking a probabilistic modeling perspective (Gardner et al., 2015b; Song et al., 2015; Gardner et al., 2015a). This approach assumes that the hearing threshold of a person is drawn from some prior probability distribution. Under this assumption, the estimation problem reduces to a (Bayesian) inference task. Since the resulting posterior distribution describes both the estimated threshold and its uncertainty, it is possible to select the 'optimal' next test tone based on information-theoretic criteria. The so-called *active learning loop* (Cohn et al., 1996) of repeatedly selecting the best next experiment and updating the probabilistic estimate, significantly reduces the total number of required test tones (Gardner et al., 2015b).

The success of the probabilistic approach hinges on the selection of a suitable hearing loss model. Presently, the Gaussian process (GP) model is the best-performing model of the hearing threshold as a function of frequency (Gardner et al., 2015b). A GP can be viewed as a probability distribution over the space of real-valued functions (Rasmussen & Williams, 2006).

In this abstract we introduce a prior distribution for hearing thresholds learned from a large database containing the hearing thresholds, ages and genders of around 85,000 people. Almost all existing work is based on very simple and/or uninformative GP priors; simply selecting a suitable type of kernel that assumes the threshold curve to be smooth is already sufficient to yield a well working system. However, by fitting a slightly more complex model to a vast database of measured thresholds, we obtain a prior that is more informative and empirically justified.

2. Probabilistic hearing loss model

The hearing threshold is a (continuous) function of frequency, denoted by $t : \mathbb{R} \to \mathbb{R}$. The goal is to specify an appropriate prior distribution $p(t|a, g)$ conditioned on age $a \in \mathbb{N}$ and gender $g \in \{\text{female}, \text{male}\}$. We choose $p(t|a, g)$ to be a Gaussian process mixture model in which the mixing weights depend on age and gender:

$$p(t|a, g) = \sum_{k=1}^{K} \pi_k(a, g)\mathcal{GP}(t|\boldsymbol{\theta}_k). \tag{1}$$

All K GPs have independent mean functions and kernels, parametrized by hyperparameter vectors $\{\boldsymbol{\theta}_k\}$. In our experiments we use third-order polynomial mean functions and the squared exponential kernel, which enforces a certain degree of smoothness on the threshold function, depending on its length-scale parameter. We do not fix mixing function $\pi(\cdot, \cdot)$ to a specific parametric form, but use a nearest neighbor regression model.

The main idea behind the choice for a mixture model is that it seems reasonable to assume that hearing thresholds can roughly be classified into several types. These types would correspond to different degrees of overall hearing loss severity, as well as hearing loss resulting from different causes, i.e. natural ageing versus extensive exposure to loud noises. The audiology literature indeed describes sets of "standard audiograms" to this end (Bisgaard et al., 2010).

3. Model fitting and evaluation

We fit the model parameters – GP hyperparameters $\boldsymbol{\theta}_1$ through $\boldsymbol{\theta}_K$ and mixing function $\pi(a, g)$ – to a database containing roughly 85k anonymized records from the Nordic countries. Each record contains the age and gender of the person in question, together with the hearing thresholds of both ears measured at (a subset of) the standard audiometric frequencies. The total set of 170k threshold measurement vectors is randomly split into a training set (80%) and a test set (20%) for performance evaluation.

The inference algorithm consists of two parts. Since all threshold measurement vectors correspond to a fixed set of frequencies, the GP mixture reduces to a mixture of multivariate Gaussians. Therefore, in the first part we fit a Gaussian mixture model to the training set using the expectation maximization algorithm (Moon, 1996). In the second part, we find the optimal GP hyperparameter values by minimizing the Kullback-Leibler divergence between the GP mixture and the multivariate Gaussian mixture using gradient descent.

Figure 2 visualizes the fitted prior conditioned on different ages. The means of the mixture components indicate that different components indeed capture different types of threshold curves. Moreover, conditioning the prior on age has a clearly visible impact. This impact is quantified in Figure 1, which shows the average log-likelihood of hearing thresholds in the test set. It also shows that the GP mixture priors outperform the empirically optimized single GP prior in terms of predictive accuracy.

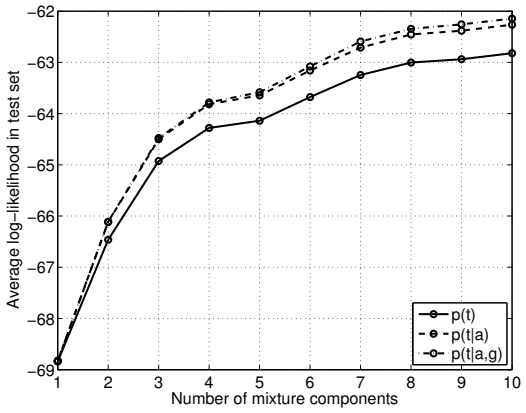
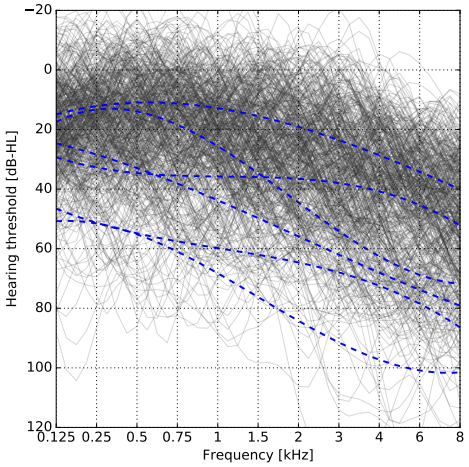
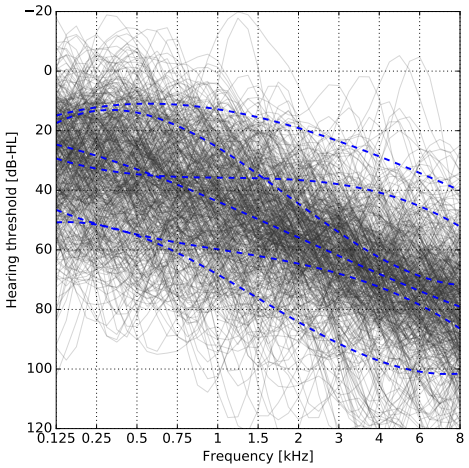


Figure 1. Predictive performance of the fitted priors on the test set. The one mixture component case corresponds to a standard GP prior with empirically optimized hyperparameters. Conditioning on age and/or gender consistently improves the predictive accuracy.



(a) $p(t|a = 40)$



(b) $p(t|a = 80)$

Figure 2. Visualization of the learned prior for $K = 6$ mixture components, conditioned on different ages. Blue dashed lines indicate the means of the mixture components. The gray lines are samples from the conditional priors. A value of 0 dB-HL corresponds to no hearing loss.

4. Conclusions

We obtained a prior for hearing loss by fitting a GP mixture model to a vast database. Evaluation on a test set shows that the mixture model outperforms the (empirically optimized) GP prior used in existing work (Gardner et al., 2015b), even without conditioning on age and gender. If age and gender are observed, the prior consistently becomes more informative. The benefit of adding more components to the mixture tapers off after about eight components.

References

Bisgaard, N., Vlaming, M. S. M. G., & Dahlquist, M. (2010). Standard audiograms for the IEC 60118-15 measurement procedure. *Trends in amplification*, *14*, 113–120.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of artificial intelligence research*, *4*, 129–145.

Gardner, J., Malkomes, G., Garnett, R., Weinberger, K. Q., Barbour, D., & Cunningham, J. P. (2015a). Bayesian active model selection with an application to automated audiometry. *Advances in Neural Information Processing Systems* (pp. 2386–2394).

Gardner, J. R., Song, X., Weinberger, K. Q., Barbour, D. L., & Cunningham, J. P. (2015b). Psychophysical Detection Testing with Bayesian Active Learning. *UAI* (pp. 286–295).

Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, *13*, 47–60.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Song, X. D., Wallace, B. M., Gardner, J. R., Ledbetter, N. M., Weinberger, K. Q., & Barbour, D. L. (2015). Fast, continuous audiogram estimation using machine learning. *Ear and hearing*, *36*, e326.

Yost, W. A. (1994). *Fundamentals of hearing: An introduction (3rd ed.)*, vol. xiii. San Diego, CA, US: Academic Press.

Predicting chaotic time series using a photonic reservoir computer with output feedback

Piotr Antonik PANTONIK@ULB.AC.BE
Laboratoire d'Information Quantique, Université libre de Bruxelles, Av. F. D. Roosevelt 50, CP 224, Brussels, Belgium

Marc Haelterman MARC.HAELTERMAN@ULB.AC.BE
Service OPERA-Photonique, Université libre de Bruxelles, Avenue F. D. Roosevelt 50, CP 194/5, Brussels, Belgium

Serge Massar SMASSAR@ULB.AC.BE
Laboratoire d'Information Quantique, Université libre de Bruxelles, Av. F. D. Roosevelt 50, CP 224, Brussels, Belgium

Keywords: Reservoir computing, opto-electronic systems, FPGA, chaotic time series prediction

1. Introduction

Reservoir Computing is a bio-inspired computing paradigm for processing time dependent signals (Jaeger & Haas, 2004; Maass et al., 2002). The performance of its hardware implementations matches digital algorithms on a series of benchmark tasks (see e.g. (Soriano et al., 2015) for a review). Their capacities could be extended by feeding the output signal back into the reservoir, which would allow them to be applied to various signal generation tasks (Antonik et al., 2016b). In practice, this requires a high-speed readout layer for real-time output computation. Here we achieve this by means of a field-programmable gate array (FPGA), and demonstrate the first photonic reservoir computer with output feedback. We test our setup on the Mackey-Glass chaotic time series generation task and obtain interesting prediction horizons, comparable to numerical simulations, with ample room for further improvement. Our work thus demonstrates the potential offered by the output feedback and opens a new area of novel applications for photonic reservoir computing. A more detailed description of this work can be found in (Antonik et al., 2017a; Antonik et al., 2017b).

2. Theory and methods

Reservoir computing. A general reservoir computer is described in (Lukoševičius & Jaeger, 2009). In our implementation we use a sine transfer function and a ring topology to simplify the interconnection matrix, so that only the first neighbour nodes are connected (Paquot et al., 2012). The system is trained offline, using ridge regression algorithm.

Mackey-Glass chaotic series generation task. The Mackey-Glass delay differential equation is given by (Mackey & Glass, 1977)

$$\frac{dx}{dt} = \beta \frac{x(t-\tau)}{1+x^n(t-\tau)} - \gamma x \qquad (1)$$

with τ, γ, β, $n > 0$. To obtain chaotic dynamics, we set the parameters as in (Jaeger & Haas, 2004): $\beta = 0.2$, $\gamma = 0.1$, $\tau = 17$ and $n = 10$. The equation was solved using the RK4 method with a stepsize of 1.0.

During the training phase, the reservoir computer receives the Mackey-Glass time series as input and is trained to predict the next value of the series from the current one. Then, the reservoir input is switched from the teacher sequence to the reservoir output signal, and the system is left running autonomously. To evaluate the system performance, we compute the number of correctly predicted steps.

3. Experimental setup

Our experimental setup, schematised in figure 1, consists of two main components: the opto-electronic reservoir and the FPGA board. The former is based on previously published works (Paquot et al., 2012). The reservoir size N depends on the delay created by the fibre spool (Spool). We performed experiments with

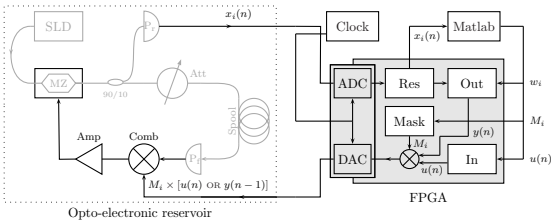


Figure 1. Schematic representation of the experimental setup. Optical and electronic components of the photonic reservoir are shown in grey and black, respectively. It contains an incoherent light source (SLD: Superluminescent Diode), a Mach-Zehnder intensity modulator (MZ), a 90/10 beam splitter, an optical attenuator (Att), a fibre spool (Spool), two photodiodes (P_r and P_f), a resistive combiner (Comb) and an amplifier (Amp). The FPGA board implements the readout layer and computes the output signal $y(n)$ in real time. It also generates the analogue input signal $u(n)$ and acquires the reservoir states $x_i(n)$ (Antonik et al., 2016a). The computer, running Matlab, controls the devices, performs the offline training and uploads all the data – inputs $u(n)$, readout weights w_i and input mask M_i – on the FPGA.

two spools of approximately 1.6 km and 10 km and, correspondingly, reservoirs of 100 and 600 neurons.

4. Results

Numerical simulations. While this work focuses on experimental results, we also developed three numerical models of the setup in order to have several points of comparison: (a) the idealised model incorporates the core characteristics of our reservoir computer, disregarding experimental considerations, and is used to define maximal achievable performance, (b) the noiseless experimental model emulates the most influential features of the experimental setup, but neglects the noise, that is taken into account by (c) the noisy experimental model.

Experimental results. The system was trained over 1000 input samples and was running autonomously for 600 timesteps. We discovered that the noise inside the opto-electronic reservoir makes the outcome of an experiment inconsistent. That is, several repetitions of the experiment with same parameters may result in significantly different prediction lengths. While the system produced several very good predictions, most of the outcomes were rather poor. We obtained similar behaviour with the noisy experimental model, using the same level of noise as measured experimentally.

Numerical simulations have shown that reducing the noise does not always increase the maximum performance, but only makes the outcome more consistent. For this reason, we measured the performances of our experimental setup by repeating the autonomous run 50 times for each training, and reporting results for the best prediction length.

Prediction length	N	
	100	600
experimental	125 ± 14	344 ± 64
numerical (noisy)	120 ± 32	361 ± 87
numerical (noiseless)	121 ± 38	637 ± 252
idealised model	217 ± 156	683 ± 264

Table 1. Summary of experimental and numerical results.

Table 1 sums up the results obtained experimentally with both reservoir sizes, as well as numerical results obtained with all three models. The prediction lengths were averaged over 10 sequences of the MG series (generated from different starting points), and the uncertainty corresponds to deviations which occurred from one sequence to another. For a small reservoir $N = 100$, experimental results agree with both experimental models, but all three are much lower than the idealised model. We found that this is due to the 23 zeroed input mask elements, as well as the limited resolution of the analog-to-digital converter (see complementary material for details). Prediction lengths obtained with the large reservoir $N = 600$ match the noisy experimental model, but here the noise has a significant impact on the maximal performance achievable.

5. Perspectives

Our numerical simulations have shown that reducing the noise inside the opto-electronic reservoir would significantly improve its performance. This can be done by upgrading the components by low noise, low voltage models, thus reducing the effects of electrical noise. Despite these issues, our work experimentally demonstrates that photonic reservoir computers are capable of emulating chaotic attractors, which offers new potential applications to this computational paradigm.

Acknowledgments

We acknowledge financial support by Interuniversity Attraction Poles program of the Belgian Science Policy Office under grant IAP P7-35 photonics@be, by the Fonds de la Recherche Scientifique F.R.S.-FNRS and by the Action de Recherche Concertée of the Académie Wallonie-Bruxelles under grant AUWB-2012-12/17-ULB9.

References

Antonik, P., Duport, F., Hermans, M., Smerieri, A., Haelterman, M., & Massar, S. (2016a). Online training of an opto-electronic reservoir computer applied to real-time channel equalization. *IEEE Transactions on Neural Networks and Learning Systems*, *PP*, 1–13.

Antonik, P., Hermans, M., Duport, F., Haelterman, M., & Massar, S. (2016b). Towards pattern generation and chaotic series prediction with photonic reservoir computers. *SPIE's 2016 Laser Technology and Industrial Laser Conference* (p. 97320B).

Antonik, P., Hermans, M., Haelterman, M., & Massar, S. (2017a). Chaotic time series prediction using a photonic reservoir computer with output feedback. *AAAI Conference on Artificial Intelligence*.

Antonik, P., Hermans, M., Haelterman, M., & Massar, S. (2017b). Photonic reservoir computer with output feedback for chaotic time series prediction. *2017 International Joint Conference on Neural Networks*. to appear.

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, *304*, 78–80.

Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Comp. Sci. Rev.*, *3*, 127–149.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural comput.*, *14*, 2531–2560.

Mackey, M. C., & Glass, L. (1977). Oscillation and chaos in physiological control systems. *Science*, *197*, 287–289.

Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., & Massar, S. (2012). Optoelectronic reservoir computing. *Sci. Rep.*, *2*, 287.

Soriano, M. C., Brunner, D., Escalona-Morán, M., Mirasso, C. R., & Fischer, I. (2015). Minimal approach to neuro-inspired information processing. *Frontiers in computational neuroscience*, *9*.

Towards high-performance analogue readout layers for photonic reservoir computers

Piotr Antonik PANTONIK@ULB.AC.BE
Laboratoire d'Information Quantique, Université libre de Bruxelles, Av. F. D. Roosevelt 50, CP 224, Brussels, Belgium

Marc Haelterman MARC.HAELTERMAN@ULB.AC.BE
Service OPERA-Photonique, Université libre de Bruxelles, Avenue F. D. Roosevelt 50, CP 194/5, Brussels, Belgium

Serge Massar SMASSAR@ULB.AC.BE
Laboratoire d'Information Quantique, Université libre de Bruxelles, Av. F. D. Roosevelt 50, CP 224, Brussels, Belgium

Keywords: Reservoir computing, opto-electronics, analogue readout, FPGA, online training

1. Introduction

Reservoir Computing is a bio-inspired computing paradigm for processing time-dependent signals (Jaeger & Haas, 2004; Maass et al., 2002). The performance of its hardware implementations (see e.g. (Soriano et al., 2015) for a review) is comparable to state-of-the-art digital algorithms on a series of benchmark tasks. The major bottleneck of these implementation is the readout layer, based on slow offline post-processing. Several analogue solutions have been proposed (Smerieri et al., 2012; Duport et al., 2016; Vinckier et al., 2016), but all suffered from noticeable decrease in performance due to added complexity of the setup. Here we propose the online learning approach to solve these issues. We present an experimental reservoir computer with a simple analogue readout layer, based on previous works, and show numerically that online learning allows to disregard the added complexity of an analogue layer and obtain the same level of performance as with a digital layer. This work thus demonstrates that online training allows building high-performance fully-analogue reservoir computers, and represents an important step towards experimental validation of the proposed solution. A more detailed description of this work can be found in (Antonik et al., 2017a; Antonik et al., 2017b).

2. Theory and methods

Reservoir computing. A general reservoir computer is described in (Lukoševičius & Jaeger, 2009). In our implementation we use a sine transfer function and a ring topology to simplify the interconnection matrix, so that only the first neighbour nodes are connected (Paquot et al., 2012). The system is trained online, using the simple gradient descent algorithm, as in (Antonik et al., 2016a).

Benchmark tasks. We tested the performance of our system on two benchmark tasks, commonly used by the RC community: wireless channel equalisation and NARMA10. The first, introduced in (Jaeger & Haas, 2004) aims at recovering the transmitted message from the output of a noisy nonlinear wireless communication channel. The performance of the equaliser is measured in terms of Symbol Error Rate (SER), that is, the number of misclassified symbols. The NARMA10 task (Atiya & Parlos, 2000) constists in emulating a nonlinear system of order 10. The performance is measured in terms of Normalised Mean Square Error (NMSE).

3. Experimental setup

Our experimental setup, which we simulate numerically, is schematised in figure 1. It consists of the opto-electronic reservoir (a replica of (Paquot et al., 2012)), the analogue readout layer, based on previous works (Smerieri et al., 2012; Duport et al., 2016), and the FPGA board, performing the online training (Antonik et al., 2016a). The readout layer uses a dual-output Mach-Zehnder modulator in order to apply both positive and negative readout weights, and the integration

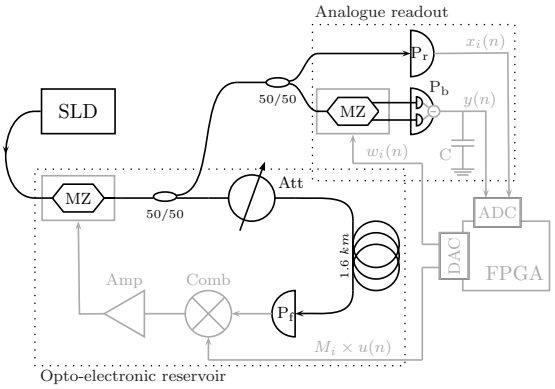


Figure 1. Scheme of the proposed experimental setup. The optical and electronic components are shown in black and grey, respectively. The reservoir layer consists of an incoherent light source (SLD), a Mach-Zehnder intensity modulator (MZ), a 50/50 beam splitter, an optical attenuator (Att), an approximately 1.6 km fibre spool, a feedback photodiode (P_f), a resistive combiner (Comb) and an amplifier (Amp). The analogue readout layer contains another 50/50 beam splitter, a readout photodiode (P_r), a dual-output intensity modulator (MZ), a balanced photodiode (P_b) and a capacitor (C). The FPGA board generates the inputs and the readout weights, samples the reservoir states and the output signal, and trains the system.

(summation) of the weighted states is carried out by a low-pass RC filter.

4. Results

All numerical experiments were performed in Matlab, using a custom model of a reservoir computer, based on previous investigations (Paquot et al., 2012; Antonik et al., 2016a).

The performance of our system on the channel equalisation task, with SERs between 10^{-4} and 10^{-3} depending on the input mask, is comparable to the same opto-electronic setup with a digital output layer (SER = 10^{-4} reported in (Paquot et al., 2012)), as well as the fully-analogue setup (Duport et al., 2016), also reporting SER of 10^{-4}. However, it outperforms the first (and, conceptually, simpler) readout layer by an order of magnitude (Smerieri et al., 2012). As for the NARMA10 task, we obtain a NMSE of 0.18. This is slightly worse than what was reported with a digital readout layer (0.168 ± 0.015 in (Paquot et al., 2012)), but better than the fully analogue setup (0.230 ± 0.023 in (Duport et al., 2016)).

Another goal of the simulations was to check how the online learning approach would cope with experimental difficulties encountered in previous works (Smerieri et al., 2012; Duport et al., 2016). To that end, we considered several potential experimental imperfection and measured their impact on the performance.

- The time constant $\tau = RC$ of the RC filter determines its integration period. We've shown that both tasks work well in a wide range of values of τ, and knowledge of its precise value is not necessary for good performance (contrary to (Duport et al., 2016)).
- The sine transfer function of the readout Mach-Zehnder modulator can, in practice, be biased due to temperature or electronic drifts of the device. This could have a detrimential impact on the readout weights. We've shown that precompensation of the transfer function is not necessary, and that realistic drifts of the bias wouldn't decrease the performance of the system.
- The numerical precision of the readout weights, limited to 16 bits by the DAC, could be insufficient for correct output generation. We've shown that resolution as low as 8 bits is enough for this application.

5. Perspectives

The present work shows that online learning allows to efficiently train an analogue readout layer despite its inherent complexity and practical imperfections. The upcoming experimental validation of this idea would lead to a fully-analogue, high-performance reservoir computer. On top of considerable speed increase, due to the removal of the slow digital post-processing, such device could be applied to periodic or chaotic signal generation by feeding the output signal back into the reservoir (Antonik et al., 2016b). This work is therefore an important step towards a new area of research in reservoir computing field.

6. Acknowledgments

We acknowledge financial support by Interuniversity Attraction Poles program of the Belgian Science Policy Office under grant IAP P7-35 photonics@be, by the Fonds de la Recherche Scientifique FRS-FNRS and by the Action de Recherche Concertée of the Académie Wallonie-Bruxelles under grant AUWB-2012-12/17-ULB9.

References

Antonik, P., Duport, F., Hermans, M., Smerieri, A., Haelterman, M., & Massar, S. (2016a). Online train-

ing of an opto-electronic reservoir computer applied to real-time channel equalization. *IEEE Transactions on Neural Networks and Learning Systems*, *PP*, 1–13.

Antonik, P., Haelterman, M., & Massar, S. (2017a). Improving performance of analogue readout layers for photonic reservoir computers with online learning. *AAAI Conference on Artificial Intelligence*.

Antonik, P., Haelterman, M., & Massar, S. (2017b). Online training for high-performance analogue readout layers in photonic reservoir computers. *Cognitive Computation*, 1–10.

Antonik, P., Hermans, M., Duport, F., Haelterman, M., & Massar, S. (2016b). Towards pattern generation and chaotic series prediction with photonic reservoir computers. *SPIE's 2016 Laser Technology and Industrial Laser Conference* (p. 97320B).

Atiya, A., & Parlos, A. (2000). New results on recurrent network training: Unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, *11*, 697–709.

Duport, F., Smerieri, A., Akrout, A., Haelterman, M., & Massar, S. (2016). Fully analogue photonic reservoir computer. *Sci. Rep.*, *6*, 22381.

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, *304*, 78–80.

Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Comp. Sci. Rev.*, *3*, 127–149.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural comput.*, *14*, 2531–2560.

Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., & Massar, S. (2012). Optoelectronic reservoir computing. *Sci. Rep.*, *2*, 287.

Smerieri, A., Duport, F., Paquot, Y., Schrauwen, B., Haelterman, M., & Massar, S. (2012). Analog readout for optical reservoir computers (pp. 944–952.).

Soriano, M. C., Brunner, D., Escalona-Morán, M., Mirasso, C. R., & Fischer, I. (2015). Minimal approach to neuro-inspired information processing. *Frontiers in computational neuroscience*, *9*.

Vinckier, Q., Bouwens, A., Haelterman, M., & Massar, S. (2016). Autonomous all-photonic processor based on reservoir computing paradigm (p. SF1F.1.). Optical Society of America.

Local Process Models: Pattern Mining with Process Models

Niek Tax, Natalia Sidorova, Wil M.P. van der Aalst {N.TAX,N.SIDOROVA,W.M.P.V.D.AALST}@TUE.NL
Eindhoven University of Technology, The Netherlands

Keywords: pattern mining, process mining, business process modeling, data mining

1. Introduction

Process mining aims to extract novel insights from event data (van der Aalst, 2016). Process discovery plays a prominent role in process mining. The goal is to discover a process model that is representative for the set of event sequences in terms of start-to-end behavior, i.e. from the start of a case till its termination. Many process discovery algorithms have been proposed and applied to a variety of real life cases. A more conventional perspective on discovering insights from event sequences can be found in the areas of sequential pattern mining (Agrawal & Srikant, 1995) and episode mining (Mannila et al., 1997), which focus on finding frequent patterns, not aiming for descriptions of the full event sequences from start to end.

Sequential pattern mining is limited to the discovery of *sequential orderings* of events, while process discovery methods aim to discover a larger set of event relations, including sequential orderings, (exclusive) choice relations, concurrency, and loops, represented in process models such as Petri nets (Reisig, 2012), BPMN (Object Management Group, 2011), or UML activity diagrams. Process models distinguish themselves from more traditional sequence mining approaches like Hidden Markov Models (Rabiner, 1989) and Recurrent Neural Networks with their visual representation, which allows them to be used for communication between process stakeholders. However, process discovery is normally limited to the discovery of a *complete* model that captures the full behavior of process instances, and not local patterns within instances. Local Process Models (LPMs) allow the mining of patterns positioned in-between simple patterns (e.g. subsequences) and end-to-end models, focusing on a subset of the process activities and describing frequent patterns of behavior.

2. Motivating Example

Imagine a sales department where multiple sales officers perform four types of activities: (A) register a call for bids, (B) investigate a call for bids from the business perspective, (C) investigate a call for bids from the legal perspective, and (D) decide on participation in the call for bid. The event sequences (Figure 1(a)) contain the activities performed by one sales officer throughout the day. The sales officer works on different calls for bids and not necessarily performs all activities for a particular call himself. Applying discovery algorithms, like the Inductive Miner (Leemans et al., 2013), yields models allowing for any sequence of events (Figure 1(c)). Such "flower-like" models do not give any insight in typical behavioral patterns. When we apply any sequential pattern mining algorithm using a threshold of six occurrences, we obtain the seven length-three sequential patterns depicted in Figure 1(d) (results obtained using the SPMF (Fournier-Viger et al., 2014) implementation of the PrefixSpan algorithm (Pei et al., 2001)). However, the data contains a frequent non-sequential pattern where a sales officer first performs A, followed by B and C in arbitrary order (Figure 1(b)). This pattern cannot be found with existing process discovery or sequential pattern mining techniques. The two numbers shown in the transitions (i.e., rectangles) represent (1) the number of events of this type in the event log that fit this local process model and (2) the total number of events of this type in the event log. For example, 13 out of 19 events of type C in the event log fit transition C, which are indicated in bold in the log in Figure 1(a). Underlined sequences indicate non-continuous instances, i.e. instances with non-fitting events in-between the events forming the instance of the local process model.

3. LPM Discovery Approach

A technique for the discovery of Local Process Models (LPMs) is described in detail in (Tax et al., 2016a). LPM discovery uses the process tree (Buijs et al., 2012) process model notation, an example of which is $SEQ(A, B)$, which is a sequential pattern that describes that activity B occurs after activity A. Process tree models are iteratively ex-

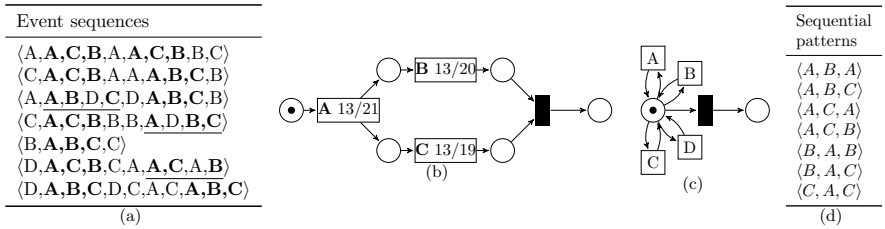


Figure 1. (a) A log L of event sequences executed by a sales officer with highlighted instances of the frequent pattern. *(b)* The local process model showing frequent behavior in L. *(c)* The Petri net discovered on L with the Inductive Miner algorithm (Leemans et al., 2013). *(d)* The sequential patterns discovered on L with PrefixSpan (Pei et al., 2001).

panded into larger patterns using a fixed set of expansion rules, e.g., $SEQ(A, B)$ can be grown into $SEQ(A, AND(B, C))$, which indicates that A is followed by both B and C in arbitrary order. Process trees can be converted in other process model notations, e.g., $SEQ(A, AND(B, C))$ can be converted in the Petri net of Figure 1(b). LPMs are discovered using the following steps:

1) **Generation** Generate the initial set CM_1 of candidate LPMs in the form of process trees.
2) **Evaluation** Evaluate LPMs in the current candidate set CM_i based on support and confidence.
3) **Selection** A subset $SCM_i \subseteq CM_i$ of candidate LPMs is selected. $SM = SM \cup SCM_i$. If $SCM_i = \emptyset$ or $i \geq max\_iterations$: stop.
4) **Expansion** Expand SCM_i into a set of larger, expanded, candidate process models, CM_{i+1}. Goto step 2 using the newly created candidate set CM_{i+1}.

4. Faster LPM Discovery by Clustering Activities

The discovery of Local Process Models (LPMs) is computationally expensive for event logs with many unique activities (i.e. event types), as the number of ways to expand each candidate LPM is equal to the number of possible process model structures with which it can be expanded times the number of activities in the log. (Tax et al., 2016b) explores techniques to cluster the set of activities, such that LPM discovery can be applied per activity cluster instead of on the complete set of events, leading to considerable speedups. All clustering techniques operate on a *directly-follows graph*, which shows how frequently the activity types of the directly follows each other in the event log. Three clustering techniques have been compared: *entropy-based clustering* clusters the activities of the directly-follows graph using an information theoretic approach. *Maximal relative information gain clustering* is a variant on entropy-based clustering. The third clustering technique uses Markov clustering (van Dongen, 2008), an out-of-the-box graph clustering technique, to cluster the activities in the directly-follows graph.

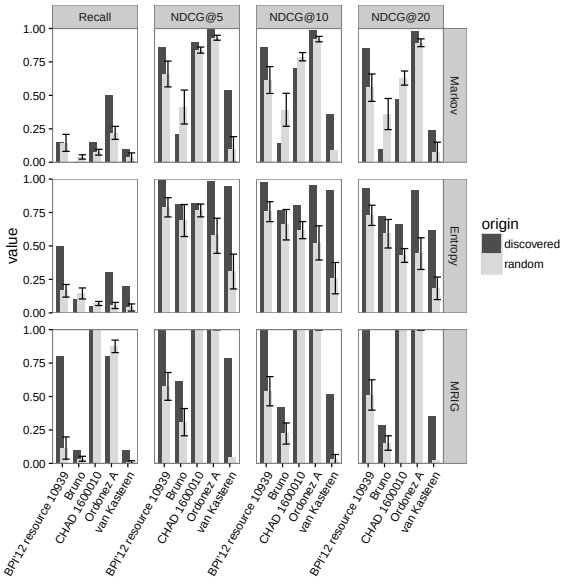


Figure 2. Performance of the three projection set discovery methods on the six data sets on the four metrics

We compare the quality of the obtained ranking of LPMs after clustering the activities with the ranking of LPMs obtained on the original data set. To compare the rankings we use NDCG, an evaluation measure for rankings frequently used in the information retrieval field. Figure 2 shows the results of the three clustering approaches on five data sets. All three produce better than random projections on a variety of data sets. Projection discovery based on Markov clustering leads to the highest speedup, while higher quality LPMs can be discovered using a projection discovery based on log statistics entropy. The Maximal Relative Information Gain based approach to projection discovery shows unstable performance with the highest gain in LPM quality over random projections on some event logs, while not being able to discover any projection smaller than the complete set of activities on some other event logs.

References

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. *Proceedings of the 11th International Conference on Data Engineering (ICDE)* (pp. 3–14). IEEE.

Buijs, J. C. A. M., van Dongen, B. F., & van der Aalst, W. M. P. (2012). A genetic algorithm for discovering process trees. *Proceedings of the 2012 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1–8).

Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani, A., Wu, C.-W., & Tseng, V. S. (2014). SPMF: a java open-source pattern mining library. *The Journal of Machine Learning Research*, *15*, 3389–3393.

Leemans, S. J. J., Fahland, D., & van der Aalst, W. M. P. (2013). Discovering block-structured process models from event logs - a constructive approach. In *Application and theory of petri nets and concurrency*, 311–329. Springer.

Mannila, H., Toivonen, H., & Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, *1*, 259–289.

Object Management Group (2011). Notation (BPMN) version 2.0. *OMG Specification*.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M.-C. (2001). PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. *Proceedings of the 17th International Conference on Data Engineering (ICDE)* (pp. 215–224).

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*, 257–286.

Reisig, W. (2012). *Petri nets: an introduction*, vol. 4. Springer Science & Business Media.

Tax, N., Sidorova, N., Haakma, R., & van der Aalst, W. M. P. (2016a). Mining local process models. *Journal of Innovation in Digital Ecosystems*, *3*, 183–196.

Tax, N., Sidorova, N., van der Aalst, W. M. P., & Haakma, R. (2016b). Heuristic approaches for generating local process models through log projections. *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining* (pp. 1–8). IEEE.

van der Aalst, W. M. P. (2016). *Process mining: Data science in action*. Springer.

van Dongen, S. (2008). Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, *30*, 121–141.

A non-linear Granger causality approach for understanding climate-vegetation dynamics

Christina Papagiannopoulou CHRISTINA.PAPAGIANNOPOULOU@UGENT.BE
Stijn Decubber STIJN.DECUBBER@UGENT.BE
Willem Waegeman WILLEM.WAEGEMAN@UGENT.BE
Depart. of Mathematical modelling, Statistics and Bioinformatics, Ghent University, Belgium

Matthias Demuzere MATTHIAS.DEMUZERE@UGENT.BE
Niko E. C. Verhoest NIKO.VERHOEST@UGENT.BE
Diego G. Miralles DIEGO.MIRALLES@UGENT.BE
Laboratory of Hydrology and Water Management, Ghent University, Belgium

Keywords: time series forecasting, random forests, non-linear Granger causality, climate change

Abstract

Satellite Earth observation provides new means to unravel the drivers of long-term changes in climate. Global historical records of crucial environmental and climatic variables, which have the form of multivariate time series, now span up to 30 years. In this abstract we present a non-linear Granger causality approach to detect causal relationships between climatic time series and vegetation. Our framework consists of several components, including data fusion from various databases, time series decomposition techniques, feature construction methods and Granger causality analysis by means of machine learning algorithms. Experimental results on large-scale entire-globe datasets indicate that, with this framework, it is possible to detect non-linear patterns that express the complex relationships between climate and vegetation.

Satellites form the only practical means for a global and continuous observation of our planet. Independent sensors on different platforms monitor the dynamics of vegetation, soils, oceans and atmosphere, collecting optical, thermal, or gravimetry information (Su et al., 2011). Their records take the form of multivariate time series with different spatial and temporal resolutions and measure environmental and climatic variables.

Vegetation plays a crucial role in the global climate system. It affects the water, the energy and the carbon cycles through the transfer of water vapor from land to atmosphere, direct effects on the surface net radiation or through the exchange of carbon dioxide with the atmosphere (McPherson et al., 2007; Bonan, 2008). Given the impact of climate on vegetation dynamics, a better understanding of the response of vegetation to projected disturbances in climatic conditions is crucial to further improve our knowledge about the potential consequences of climate change. A first and necessary step in this direction, however, is to investigate the response of vegetation to past-time climate variability.

Simple correlation statistics and linear regression methods in general are insufficient when it comes to assessing causality, especially in high dimensional datasets and given the non-linear nature of climate–vegetation dynamics. A commonly used approach consists of Granger causality modelling (Granger, 1969), which is typically expressed through linear vector autoregressive (VAR) models. In Granger causality, it is assumed that a time series x (in our case, climatic time series) Granger-causes another time series y (i.e., vegetation) if the past of x is helpful in predicting the future of y, given the past of y. In practice, the forecasting accuracy for future values of y of two competing models is compared: a *full* model which includes both past-time y and x as predictors and a purely autoregressive *baseline* model which just has access to past-time y. If the full model produces significantly

Appearing in *Proceedings of Benelearn 2017*.

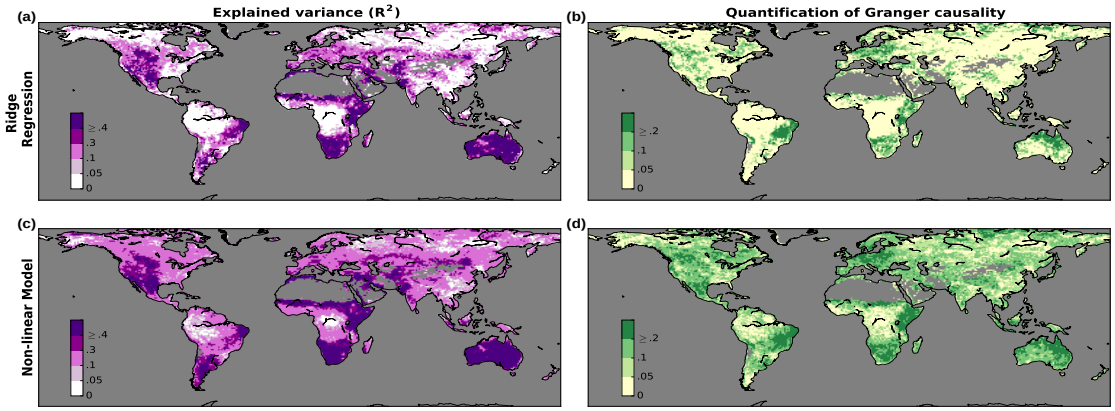


Figure 1. Linear versus non-linear Granger causality of climate on vegetation. (a) Explained variance (R^2) of vegetation anomalies based on a full ridge regression model in which all climatic variables are included as predictors. (b) Improvement in terms of R^2 by the full ridge regression model with respect to the baseline ridge regression model that uses only past values of vegetation anomalies as predictors; positive values indicate (linear) Granger causality. (c) Explained variance (R^2) of vegetation anomalies based on a full random forest. (d) Improvement in terms of R^2 by the full random forest model with respect to the baseline random forest model; positive values indicate (non-linear) Granger causality.

better forecasts, the null hypothesis of Granger non-causality can be rejected (Granger, 1969).

This abstract, based on (Papagiannopoulou et al., 2016), presents an extension of linear Granger causality analysis, a novel non-linear framework for finding climatic drivers that affect vegetation. Our framework consists of several steps. In a first step, data from different sources are collected and merged into a single, comprehensive dataset. Next, time series decomposition techniques are applied to the target vegetation time series and the various predictor climatic time series to isolate seasonal cycles, trends and anomalies. In a third step, we explore various techniques for constructing high-level features from climatic time series using techniques that are similar to shapelets (Ye & Keogh, 2009). In a final step, we run a Granger causality analysis on the vegetation anomalies, while replacing traditional linear vector autoregressive models with random forests.

Applying the above framework, we end up with 4,571 features generated on thirty-year time series, allowing to analyze 13,097 land pixels independently. Predictive performance is assessed by means of five-fold cross-validation using the out-of-sample coefficient of determination (R^2) as a performance measure.

Figure 1a shows the predictive performance of a ridge regression model which includes the 4,571 climate predictors on top of the history of vegetation (i.e., a *full model*). While the model explains more than 40% of the variability in vegetation in some regions ($R^2 > 0.4$), this is by itself not necessarily indicative of climate Granger-causing the vegetation anomalies. In order to test the latter, we compare the results of the full model (Fig. 1a) to a baseline model, i.e., an autoregressive ridge regression model that only uses previous values of vegetation to predict the vegetation at time t. Any increase in predictive performance provided by the full ridge regression model (Fig. 1a) over the corresponding baseline provides qualitative evidence of Granger causality (Fig. 1b). The results show that, when only linear relationships between vegetation and climate are considered, the areas in which Granger causality of climate towards vegetation is suggested are limited. The predictive power for vegetation anomalies increases dramatically when using random forests (Fig. 1c). In order to test whether the climatic and environmental controls Granger-cause the vegetation anomalies, we again compare the results of a full random forest model to a baseline random forest model. As seen in Fig. 1d, the improvement over the baseline is unambiguous. One can conclude that, while not bearing into consideration all potential control variables in our analysis, climate dynamics indeed Granger-cause vegetation anomalies in most of the continental land surface. Moreover, the improved capacity of random forests over ridge regression to predict vegetation anomalies suggests that these relationships are non-linear.

References

Bonan, G. (2008). Forests and climate change: forcings, feedbacks, and the climate benefits of forests. *science*, *320*, 1444–1449.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, 424–438.

McPherson, R. A., Fiebrich, C. A., Crawford, K. C., Kilby, J. R., Grimsley, D. L., Martinez, J. E., Basara, J. B., Illston, B. G., Morris, D. A., Kloesel, K. A., et al. (2007). Statewide monitoring of the mesoscale environment: A technical update on the Oklahoma Mesonet. *24*, 301–321.

Papagiannopoulou, C., Miralles, D. G., Decubber, S., Demuzere, M., Verhoest, N. E. C., Dorigo, W. A., & Waegeman, W. (2016). A non-linear Granger causality framework to investigate climate–vegetation dynamics. *Geoscientific Model Development Discussions*, *2016*, 1–24.

Su, L., Jia, W., Hou, C., & Lei, Y. (2011). Microbial biosensors: a review. *Biosensors and Bioelectronics*, *26*, 1788–1799.

Ye, L., & Keogh, E. (2009). Time series shapelets: a new primitive for data mining. *Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09* (p. 947). New York, New York, USA: ACM Press.

Characterizing Resting Brain Activity to Predict the Amplitude of Pain-Evoked Potentials in the Human Insula

Dounia Mulders, Michel Verleysen {NAME.SURNAME}@UCLOUVAIN.BE
ICTEAM institute, Université catholique de Louvain, Place du Levant 3, 1348 Louvain-la-Neuve, Belgium

Giulia Liberati, André Mouraux {NAME.SURNAME}@UCLOUVAIN.BE
IONS institute, Université catholique de Louvain, Avenue Mounier 53, 1200 Woluwe-Saint-Lambert, Belgium

Keywords: Pain, nociception, intracerebral recordings, feature extraction, time series prediction.

Abstract

How the perception of pain emerges from human brain activity remains largely unknown. Apart from inter-individual variations, this perception depends not only on the physical characteristics of the painful stimuli, but also on other psycho-physiological aspects. Indeed a painful stimulus applied to an individual can sometimes evoke very distinct sensations from one trial to the other. Hence the state of a subject receiving such a stimulus should (at least partly) explain the intensity of pain elicited by that stimulus. Using intracranial electroencephalography (iEEG) from the insula to measure this cortical "state", our goal is to study to which extent ongoing brain activity in the human insula, an area thought to play a key role in pain perception, may predict the magnitude of pain-evoked potentials and, more importantly, whether it may predict the perception intensity. To this aim, we summarize the ongoing insular activity by defining frequency-dependent features, derived using continuous wavelet and Fourier transforms. We then take advantage of this description to predict the amplitude of the insular responses elicited by painful (heat) and non-painful (auditory, visual and vibrotactile) stimuli, as well as to predict the intensity of perception.

1. Introduction

The ability to perceive pain is crucial for survival, as exemplified by the injuries and reduced life expectancy of people with congenital insensitivity to pain. Furthermore, pain is a major healthcare issue and its treatment, especially in the context of pathological chronic pain, constitutes a very challenging problem for physicians. Characterizing the relationship between pain perception and brain activity could provide insights on how nociceptive inputs are processed in the human brain and, ultimately, how this leads to the perception of pain (Apkarian et al., 2005; Tracey & Mantyh, 2007). It is widely accepted that perception fluctuates along time, even in a resting state. These fluctuations might result from variations in neuronal activity (Sadaghiani et al., 2010) which can be, at least partly, recorded with neuroimaging or electrophysiological monitoring techniques (VanRullen et al., 2011). Hence spontaneous brain activity, which is often considered as noise, might be related to our perception capabilities. However the potential links between perception fluctuations and the recorded brain activity are not yet fully elucidated. It has already been suggested that perception is a discrete process, namely that the cortex quickly oscillates between different, relatively short-lasting levels of excitability (VanRullen & Koch, 2003). This excitability can for instance be measured by electroencephalography (EEG).

Supporting the aforementioned hypothesis, several studies have already established links between ongoing brain activity measured before the presentation of a sensory stimulus using functional magnetic resonance imaging (fMRI) or EEG, and the subsequent stimulus-evoked response, assessed either in terms of subjective perception or brain response magnitude (Mayhew et al., 2013; Monto et al., 2008). For instance, Barry et al. and Busch et al. study the effect of pre-stimulus low-frequency (between 5 and 15Hz) phase on auditory and visual perception respectively, showing that stimulus processing can be affected by such phase (i.e. position within a cycle) at the stimulus onset (2004; 2009). Tu et al. use linear regression to predict the sub-

Appearing in *Proceedings of Benelearn 2017*.

jective pain intensity generated by nociceptive stimuli from the time-frequency coefficients obtained by a short-time Fourier transform (2016). They show that pain perception depends on some pre-stimulus time-frequency features.

In this setting, our work aims to study whether and to which extent pain perception capabilities vary along time. As a first step, our goal is to predict the amplitude of the responses recorded in the insula following a painful heat stimulus (generated by a CO_2 laser) from the pre-stimulus insular activity. We focus on the insula because this region is thought to play an important role in pain perception (Garcia-Larrea & Peyron, 2013). Instead of analyzing the relationships between the elicited responses and only a few features (e.g. the phase or power in one frequency band) of the resting EEG prior to a stimulus onset, we propose to first characterize this ongoing activity in more details. We summarize the ongoing EEG activity by defining frequency-dependent features, derived using continuous wavelet and Fourier transforms. This description is then exploited to predict the amplitude of the insular potentials elicited by painful (heat) and non-painful (auditory, visual and vibrotactile) stimuli (for comparison) using multilayer perceptron neural networks, linear regression and k-nearest neighbor regression.

2. Method

This study has been approved by the local ethic committee (CEBHF). Benefiting from the high spatiotemporal resolution of iEEG recordings from the insula performed in patients implanted for a presurgical evaluation of focal epilepsy, our first goal is to summarize the ongoing insular activity prior to a stimulus application. For this purpose and as oscillations in different frequency bands have been associated to different functions, frequency-dependent features are defined. Because neural activity is non-stationary and since brain functional state has been hypothesized to vary over small time scales (less than one second) (Britz et al., 2010), we propose to extract oscillation features as close as possible to the stimulus onset, while avoiding border effects specifically to each frequency band. The extracted features consist in (1) the amplitude, (2) the phase and (3) the power in the five physiological frequency bands (Birjandtalab et al., 2016). These bands are denoted by δ, θ, α, β and γ and correspond to the frequency ranges $[0.1, 4]$, $[4, 8]$, $[8, 12]$, $[12, 30]$ and $[30, 80]$ Hz. The two first kinds of features are defined using the Morlet (continuous) wavelet, allowing to describe the phase and amplitude of the oscillations at a particular time with a better temporal resolution

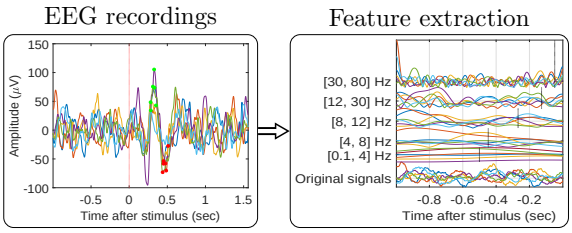


Figure 1. Extraction of the amplitude features in five frequency bands for six presentations of a painful stimulus to a subject. The first box shows the recorded signals, with the red and green dots allowing to define the response amplitude. The second box gives the continuous wavelet transforms of these trials, the vertical dotted black lines indicating the feature extraction times. These are defined by taking the wavelet support into account, according to the frequency band considered.

at higher frequencies. The power features take a larger time window into account, starting 0.5 second before the stimulus onset, as these are obtained by Fourier transform. Figure 1 shows an example of the amplitudes extraction for six trials, for which the stimulus onset is at $t = 0$.

Using the aforementioned features to describe the ongoing activity before the presentation of each stimulus (40 trials of each kind of stimulus were conducted on each patient), we then predict the response amplitude. So far, the achieved performances are not significant, but further investigations are carried out.

3. Conclusion

In line with previous works attempting to study how fluctuations of the ongoing oscillatory brain activity may modulate pain perception, our approach aims to establish a link between the combination of several features of the ongoing insular activity and the subsequent neural response, taking advantage of the high spatiotemporal resolution of intracerebral EEG. The same characterization of the spontaneous brain activity could be used to predict subject evaluated pain perception directly, rather than the observed neural response.

Acknowledgments

DM is a Research Fellow of the Fonds de la Recherche Scientifique - FNRS.

References

Apkarian, A. V., Bushnell, M. C., Treede, R.-D., & Zubieta, J.-K. (2005). Human brain mechanisms of pain perception and regulation in health and disease. *European journal of pain*, *9*, 463–463.

Barry, R. J., Rushby, J. A., Johnstone, S. J., Clarke, A. R., Croft, R. J., & Lawrence, C. A. (2004). Event-related potentials in the auditory oddball as a function of eeg alpha phase at stimulus onset. *Clinical Neurophysiology*, *115*, 2593–2601.

Birjandtalab, J., Pouyan, M. B., & Nourani, M. (2016). Nonlinear dimension reduction for eeg-based epileptic seizure detection. *Biomedical and Health Informatics (BHI), 2016 IEEE-EMBS International Conference on* (pp. 595–598).

Britz, J., Van De Ville, D., & Michel, C. M. (2010). Bold correlates of eeg topography reveal rapid resting-state network dynamics. *Neuroimage*, *52*, 1162–1170.

Busch, N. A., Dubois, J., & VanRullen, R. (2009). The phase of ongoing eeg oscillations predicts visual perception. *Journal of Neuroscience*, *29*, 7869–7876.

Garcia-Larrea, L., & Peyron, R. (2013). Pain matrices and neuropathic pain matrices: a review. *PAIN®*, *154*, S29–S43.

Mayhew, S. D., Hylands-White, N., Porcaro, C., Derbyshire, S. W., & Bagshaw, A. P. (2013). Intrinsic variability in the human response to pain is assembled from multiple, dynamic brain processes. *Neuroimage*, *75*, 68–78.

Monto, S., Palva, S., Voipio, J., & Palva, J. M. (2008). Very slow eeg fluctuations predict the dynamics of stimulus detection and oscillation amplitudes in humans. *Journal of Neuroscience*, *28*, 8268–8272.

Sadaghiani, S., Hesselmann, G., Friston, K. J., & Kleinschmidt, A. (2010). The relation of ongoing brain activity, evoked neural responses, and cognition. *Frontiers in systems neuroscience*, *4*, 20.

Tracey, I., & Mantyh, P. W. (2007). The cerebral signature for pain perception and its modulation. *Neuron*, *55*, 377–391.

Tu, Y., Zhang, Z., Tan, A., Peng, W., Hung, Y. S., Moayedi, M., Iannetti, G. D., & Hu, L. (2016). Alpha and gamma oscillation amplitudes synergistically predict the perception of forthcoming nociceptive stimuli. *Human brain mapping*, *37*, 501–514.

VanRullen, R., Busch, N., Drewes, J., & Dubois, J. (2011). Ongoing eeg phase as a trial-by-trial predictor of perceptual and attentional variability. *Frontiers in psychology*, *2*, 60.

VanRullen, R., & Koch, C. (2003). Is perception discrete or continuous? *Trends in cognitive sciences*, *7*, 207–213.

Probabilistic Inference-based Reinforcement Learning

Quan Nguyen[1] M.NGUYEN@TUE.NL
Bert de Vries[1,2] BDEVRIES@GNRESOUND.COM
Tjalling J. Tjalkens[1] T.J.TJALKENS@TUE.NL
[1]Department of Electrical Engineering, Eindhoven University of Technology, [2]GN Hearing BV, Eindhoven

Keywords: reinforcement learning, reward functions, probabilistic modeling, Bayesian inference.

Abstract

We introduce probabilistic inference-based reinforcement learning (PIReL), an approach to solve decision making problems by treating them as probabilistic inference tasks. Unlike classical reinforcement learning, which requires explicit reward functions, in PIReL they are implied by probabilistic assumptions of the model. This would enable a fundamental way to design the reward function by model selection as well as bring the potential to apply existing probabilistic modeling techniques to reinforcement learning problems.

1. Introduction

Reinforcement learning (RL) is a domain in machine learning concerning with how an *agent* makes decisions in an uncertain *environment*. In the traditional approach, the agent learns how to do a certain task by maximizing the expected total rewards. However, the reward functions are often handcrafted for specific problems than based on a general guideline.

In contrast to classical RL, probabilistic inference-based reinforcement learning (PIReL) treats the action as a hidden variable in a probabilistic model. Hence choosing actions that lead to the desired *goal states* can be treated in a straightforward manner as probabilistic inference.

This idea was in fact first proposed by (Attias, 2003). Our contribution is to extend the original framework so that it can take into account uncertainties about the goals. The extended framework shows its connection to classical RL. Particularly the reward function and discount factor in classical RL can be seen as certain probabilistic assumptions in the model. This interpretation provides us with a way to design appropriate reward function, by e.g., *model selection*.

Appearing in *Proceedings of Benelearn 2017*.

2. Problem Modeling

The model is based on the Markov Decision Process. The interaction between the agent and the environment occurs in a time sequence, so subscription is used to indicate the time step. Under the Markov assumption, when the environment is in state s_t (which is supposed to be fully observed by the agent), receives an action a_t from the agent, will change to a new state s_{t+1}. The generative model is specified as:

$$p(s_{1:T}, a_{1:T}) = \pi * p(s_1) \prod_{t=1}^{T-1} p(s_{t+1}|s_t, a_t), \quad (1)$$

where $\pi \triangleq \prod_{t=1}^{T-1} p(a_t)$ is the action prior (prior policy), and $p(s_{t+1}|s_t, a_t)$ is the *transition probability*. Unlike the standard MDP, there is no explicit reward here. Next we will explain how to infer actions.

2.1. Reinforcement Learning by Goal-based Probabilistic Inference

For the simplest decision making problem (Attias, 2003), at the initial state s_1, given a fixed horizon $T > 1$, and action prior π, the agent decides which actions $a_{1:T-1}$ should be done in order to archive the specified goal at the horizon, $s_T = g$. In the other words, we are interested in the posterior:

$$p(a_t|s_1, s_T = g), \forall t \in \{1, \ldots, T-1\}, \quad (2)$$

These probabilities have the form of a *smoothing distribution*, and the inference problem can be solved efficiently by a *forward-backward*-based algorithm.

3. Bayesian Policy and Relation to Classical Reinforcement Learning

In practice, it could be tricky to specify a desired goal precisely on s_T. Thus we introduce an abstract random binary variable z that indicates whether s_T is a good (rewarding) or bad state. The goal is instead set as $z = 1$ (good state).

In the special case when the given goal on s_T is certain, we have $p(z = 1|s_T) \triangleq \delta(s_T - g)$. And one could verify that

$$p(z = 1|s_1, T) = p(s_T = g|s_1, T)$$

while the updated policy (posterior) becomes

$$p(a_t|s_1, z = 1, T) = p(a_t|s_1, s_T = g), \forall t.$$

For an uncertain goal on s_T, we have a generic form $\pi_{zT} \triangleq p(z = 1|s_T)$, which is a probability function with input s_T (since z is always fixed at 1).

The current policy however still assumes that the horizon is known. Similarly, to accommodate the uncertainty about the horizon, we average over it. Without loss of generality, assume that horizon T is upper bounded by $1 < \mathcal{T} < \infty$, thus we have the full Bayesian policy

$$p(a_t|s_1, z = 1; \pi_T) = \sum_{T=2}^{\mathcal{T}} \pi_T p(a_t|s_1, z = 1, T), \forall t, \quad (3)$$

where $\pi_T \triangleq p(T)$ is the probability that the horizon is at time T. The marginal likelihood under $\pi_{ag} \triangleq \{\pi, \pi_T, \pi_{zT}\}$ (policy, horizon, and goal distribution, respectively) is defined as:

$$\begin{aligned} p(z = 1|s_1; \pi_{ag}) &= \sum_{T=2}^{\mathcal{T}} \pi_T \, p(z = 1|s_1, T; \pi_{zT}; \pi) \\ &= \sum_{T=2}^{\mathcal{T}} \pi_T \int \pi_{zT} \prod_{t=1}^{T-1} p(s_{t+1}|s_t; \pi) \, \mathrm{d}s_{2:T} \end{aligned} \quad (4)$$

Let's consider the *value function*, the expected (discounted) total reward up to $\mathcal{T}$, when the agent at initial state s_1 follows policy π (Sutton & Barto, 2017):

$$\begin{aligned} V_\pi(s_1) &= \mathbb{E}\left[\left(\sum_{T=1}^{\mathcal{T}} \gamma_T r_T\right)\middle| s_1; \pi\right] \\ &= \sum_{T=1}^{\mathcal{T}} \gamma_T \mathbb{E}(r_T|s_1; \pi) \\ &= \sum_{T=1}^{\mathcal{T}} \gamma_T \int R(s_T) \prod_{t=1}^{T-1} p(s_{t+1}|s_t; \pi) \, \mathrm{d}s_{2:T}, \end{aligned}$$

where γ_T and r_T denote the discount factor and instant reward at time T respectively, while $R(s_T)$ is the reward function that returns a corresponding reward for state s_T.

It is clear that the horizon distribution π_T behaves like the discount factor, while the goal distribution π_{zT} acts like the reward function in classical reinforcement learning. In classical RL, both reward function and discount factor are often given. In contrast in our probabilistic framework, the optimal policy, horizon and goal distribution $\hat{\pi}_{ag}$ that maximize the (log) marginal likelihood in eq. (4) can be estimated by e.g. EM algorithm (Dempster et al., 1977).

4. Related Work

The basic idea of PIReL originates from (Attias, 2003), where the agent infers actions in order to reach a certain goal at a fixed horizon. (Toussaint & Storkey, 2006) define the goal as to obtain the highest valued reward at the horizon; and propose an EM-based algorithm to derive the MAP estimation of action posterior with the horizon is marginalized out. By averaging over the horizons, the inferred policy also maximizes the expected return.

In the neuroscience and cognitive sciences literature, similar ideas to PIReL have been suggested, e.g., (Friston, 2010) and (Botvinick & Toussaint, 2012) discuss agents that infer actions that lead to a predefined goal.

An alternative approach to improve the reward function is *reward shaping*, see e.g. (Ng et al., 1999), which however offers a limited alteration to the predefined rewards.

5. Conclusions

We discussed a framework where classical RL is recast as goal-based probabilistic inference. In this approach, there are no explicit reward functions as in classical RL, but instead the agent infers what actions to be do in order to reach a set of goals with different priorities. The reward function and discount factor can be interpreted as the goal and horizon distribution in this probabilistic framework. This potentially brings fundamental ways to improve or design an appropriate reward function and discount factor.

Acknowledgments

This work is part of the research programme HearScan with project number 13925, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO). We thank the anonymous reviewers

for their thoughtful comments and suggestions.

References

Attias, H. (2003). Planning by probabilistic inference. *AISTATS*.

Botvinick, M., & Toussaint, M. (2012). Planning as inference. *Trends in Cognitive Sciences*, *16*, 485–488.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1–38.

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, *11*, 127–138.

Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. *International Conference on Machine Learning* (pp. 278–287).

Sutton, R., & Barto, A. (2017). *Reinforcement learning: An introduction*. The MIT Press. second (in progress) edition.

Toussaint, M., & Storkey, A. (2006). Probabilistic inference for solving discrete and continuous state Markov Decision Processes. *Proceedings of the 23rd International Conference on Machine Learning* (pp. 945–952).

Identifying Subject Experts through Clustering Analysis

Veselka Boeva VESELKA.BOEVA@BTH.SE
Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden

Milena Angelova MANGELOVA@TU-PLOVDIV.BG
Technical University of Sofia - branch Plovdiv, 4000 Plovdiv, Bulgaria

Elena Tsiporkova ELENA.TSIPORKOVA@SIRRIS.BE
Sirris, The Collective Center for the Belgian technological industry, Brussels, Belgium

Keywords: data mining, expert finding, formal concept analysis, health science, knowledge management

Abstract

In this work, we discuss an approach for identifying subject experts via clustering analysis of the available online information. Initially, the domain of interest is partitioned to a number of subject areas. Next each extracted expert is represented by a vector of contributions of the expert to the different areas. Finally, the set of all extracted experts is grouped into a set of disjoint expert areas by applying Formal Concept Analysis (FCA). The produced grouping is further shown to facilitate the location of experts with the required competence.

1. Introduction

Expertise retrieval has already gained significant interest in the area of information retrieval. A variety of practical scenarios of organizational situations that lead to expert seeking have been extensively presented in the literature, e.g., see (Cohen et al., 1998), (Kanfer et al., 1997), (Kautz et al., 1996), (Mattox et al., 1999), (McDonald & Ackerman, 1998), (Vivacqua, 1999).

In the recent years, research on identifying experts from online data sources has been gradually gaining interest (Abramowicz, 2011), (Bozzon et al., 2013), (Balog, 2007), (Hristoskova et al., 2013), (Jung et al., 2007), (Stankovic et al., 2011), (Singh et al., 2013), (Tsiporkova & Tourwè, 2011), (Zhang et al., 2007). In (Boeva et al., 2016), an enhanced technique that identifies subject experts via clustering analysis of the available online information has been developed. The authors propose a formal concept analysis approach for grouping a given set of experts with respect to pre-defined subject areas. The proposed approach is especially useful for modelling of two-phase expert identification processes. Such processes are dealing with two levels of complexity: initially it is necessary to identify a wide set of available experts, who have an expertise in a domain of interest given at the higher-level of abstraction (e.g., health science); then at the second phase the expert domain can be broken up into more specific expert areas and the available experts can further be reduced to a smaller group of experts who all have expertise in a number of selected areas (e.g., information science and health care). This expert identification scenario could be useful, e.g., when the set of available experts is preliminary identified and known, then the expert finding system can help in recruiting currently needed individuals by taking into account the specified topics.

2. Clustering of Experts

The data needed for constructing the expert profiles could be extracted from various Web sources, e.g., LinkedIn, the DBLP library, Microsoft Academic Search, Google Scholar Citation, PubMed etc. There exist several open tools for extracting data from public online sources. In addition, the Stanford part-of-speech tagger (Toutanova, 2000) can be used to annotate the different words in the text collected for each expert with their specific part of speech. The annotated text can be reduced to a set of keywords by removing all the words tagged as articles, prepositions,

Preliminary work. Under review for Benelearn 2017. Do not distribute.

verbs, and adverbs.

In view of the above, we define an expert profile as a list of keywords, extracted from the available information about the expert in question, describing her/his subjects of expertise. Assume that n different expert profiles are created in total and each expert profile i $(i = 1, 2, \ldots, n)$ is represented by a list of p_i keywords.

A conceptual model of the domain of interest, such as a thesaurus, a taxonomy etc., can be available and used to attain accurate and topic relevant expert profiles. In this case, usually a set of subject terms (topics) arranged in hierarchical manner is used to represent concepts in the considered domain. Another possibility to represent the domain of interest at a higher level of abstraction is to partition the set of all different keywords used to define the expert profiles into k main subject areas. The latter idea has been proposed and applied in (Boeva et al., 2014).

As discussed above, the domain of interest can be presented by k main subject categories $C_1, C_2, \ldots, C_k$. Let us denote by b_{ij} the number of keywords from the expert profile of expert i that belong to category C_j. Now each expert i can be represented by a vector $e_i = (e_{i1}, e_{i2}, \ldots, e_{ik})$, where $e_{ij} = b_{ij}/p_i$ and p_i is the total number of keywords in the expert profile representation. In this way, each expert i is represented by a k-length vector of membership degrees of the expert to k different subject categories, i.e. the above procedure generates a fuzzy clustering. The resulting fuzzy partition can easily be turned into a crisp one by assigning to each pair (expert, area) a binary value (0 or 1), i.e. for each subject area we can associate those experts who have membership degrees greater than a preliminary given threshold (e.g., 0.5). This partition is not guaranteed to be disjoint in terms of the different subject areas, since there will be experts who will belong to more than one subject category. This overlapping partition is further analyzed and refined into a disjoint one by applying FCA.

Formal concept analysis (Ganter et al., 2005) is a mathematical formalism allowing to derive a concept lattice from a formal context constituted of a set of objects, a set of attributes, and a binary relation defined on the Cartesian product of these two sets. In our case, a (formal) context consists of the set of the n experts, the set of main categories $\{C_1, C_2, \ldots, C_k\}$ and an indication of which experts are associated with which subject category. Thus the context is described as a matrix, with the experts corresponding to the rows and the categories corresponding to the columns of the matrix, and a value 1 in cell (i, j) whenever expert i is associated with subject area C_j. Subsequently, a concept for this context is defined to be a pair (X, Y) such that X is a subset of experts and Y is a subsets of subject areas, and every expert in X belongs to every area in Y; for every expert that is not in X, there is a subject area in Y that does not contain that expert; for every subject area that is not in Y, there is a expert in X who is not associated with that area. The family of these concepts obeys the mathematical axioms defining a concept lattice. The built lattice consists of concepts where each one represents a subset of experts belonging to a number of subject areas. The set of all concepts partitions the experts into a set of disjoint expert areas. Notice that the above introduced grouping of experts can be performed with respect to any set of subject areas describing the domain of interest, e.g., the experts could be clustered on a lower level of abstraction by using more specific topics.

3. Initial Evaluation and Discussion

The proposed approach has initially been evaluated in (Boeva et al., 2016) by applying the algorithm to partition Bulgarian health science experts extracted from PubMed repository of peer-reviewed biomedical articles. Medical Subject Headings (MeSH) is a controlled vocabulary developed by the US National Library of Medicine for indexing research publications, articles and books. Using the MeSH terms associated with peer-reviewed articles published by Bulgarian authors and indexed in the PubMed, we extract all such authors and construct their expert profiles. The MeSH headings are grouped into 16 broad subject categories. We have produced a grouping of all the extracted authors with respect to these subject categories by applying the discussed formal concept analysis approach. The produced grouping of experts is shown to capture well the expertise distribution in the considered domain with respect to the main subjects. In addition, it facilitates the identification of individuals with the required competence. For instance, if we need to recruit researchers who have expertise simultaneously in 'Phenomena and Processes' and 'Health care' categories, we can directly locate those who belong to the concept that unites the corresponding categories.

4. Conclusion and Future Work

A formal concept analysis approach for clustering of a group of experts with respect to given subject areas has been discussed. The initial evaluation has demonstrated that the proposed approach is a robust clustering technique that is suitable to deal with sparse data. Further evaluation and validation on richer data extracted from different online sources are planned.

References

Abramowicz, W. (2011). Semantically enabled experts finding system - ontologies, reasoning approach and web interface design. *Proceedings of ADBIS* (pp. 157–166).

Balog, K. (2007). Broad expertise retrieval in sparse data environments. *Proceedings of 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval ACM Press, New York.*

Boeva, V., Angelova, M., Boneva, L., & Tsiporkova, E. (2016). Identifying a group of subject experts using formal concept analysis. *Proceedings of the 8th IEEE International Conference on Intelligent Systems.*

Boeva, V., Boneva, L., & Tsiporkova, E. (2014). Semantic-aware expert partitioning. *Artificial Intelligence: Methodology, Systems, and Applications, LNAI Springer International Publishing Switzerland*, *8722*, 13–24.

Bozzon, A., Brambilla, M., Ceri, S., Silvestri, M., & Vesci, G. (2013). Choosing the right crowd: Expert finding in social networks. *Proceeding of EDBT/ICDT13, Genoa, Italy.*

Cohen, A. L., Maglio, P. P., & Barrett, R. (1998). The expertise browser: how to leverage distributed organizational knowledge. *Proceedings of CSCW98, Seatle, WA.*

Ganter, B., Stumme, G., & Wille, R. (2005). Formal concept analysis: Foundations and applications. *LNAI no. 3626 Springer-Verlag.*

Hristoskova, A., Tsiporkova, E., Tourwè, T., Buelens, S., Putman, M., & Turck, F. D. (2013). A graph-based disambiguation approach for construction of an expert repository from public online sources. *Proceeding of 5th IEEE International Conference on Agents and Artificial Inteligence.*

Jung, H., Lee, M., Kang, I.-S., Lee, S.-W., & Sung, W.-K. (2007). Finding topic-centric identified experts based on full text analysis. *Procing of 2nd International ExpertFinder Workshop, ISWC.*

Kanfer, A., Sweet, J., & Schlosser, A. (1997). Humanizing the net: social navigation with a know-who email agent. *Proceedings of the Third Conference on Human Factors and The web. Denver, Colorado.*

Kautz, H., Selman, B., & Milewski, A. (1996). Agent amplified communication. *Proceedings of the Thirteenth National Conference on Artificial Intelligence, Portland, OR.*

Mattox, D., Maybury, M., & Morey, D. (1999). Enterprise expert and knowledge discovery. *Proceedings of HCI International99, Special Session on: Computer Supported Communication and Cooperation Making Information Aware, Munich, Germany.*

McDonald, D., & Ackerman, M. (1998). Just talk to me: a field study of expertise location. *Proceedings of CSCW98, ACM Press. Seattle, WA.*

Singh, H., Singh, R., Malhotra, A., & Kaur, M. (2013). Developing a biomedical expert finding system using medical subject headings. *Healthcare Informatics Research*, *19*, 243–249.

Stankovic, M., Jovanovic, J., & Laublet, P. (2011). Linked data metrics for flexible expert search on the open web. *LNCS, Springer, Heidelberg*, 108123.

Toutanova, K. (2000). Enriching the knowledge sources used in a maximum entropy partofspeech tagger. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora EMNLP/VLC-2000.*

Tsiporkova, E., & Tourwè, T. (2011). Tool support for technology scouting using online sources. *LNCS, Springer, Heidelberg*, *6999*, 371376.

Vivacqua, A. (1999). Agents for expertise location. *Proceedings of the AAAI Spring Symp. on Intelligent Agents in Cyberspace, Stanford, CA.*

Zhang, J., Tang, J., & Li, J. (2007). Expert finding in a social network. *LNCS, Springer, Heidelberg*, 10661069.

An Exact Iterative Algorithm for Transductive Pairwise Prediction

Michiel Stock MICHIEL.STOCK@UGENT.BE
Bernard De Baets BERNARD.DEBAETS@UGENT.BE
Willem Waegeman WILLEM.WAEGEMAN@UGENT.BE
KERMIT, Coupure links 653, Ghent, Belgium

Keywords: pairwise learning, transductive learning, matrix imputation, optimization

Abstract

Imputing missing values of a matrix when side-features are available can be seen as a special case of pairwise learning. In this extended abstract we present an exact iterative algorithm to impute these missing values efficiently.

1. Problem statement

Consider the problem of pairwise learning where for a given dyad $(u, v) \in \mathcal{U} \times \mathcal{V}$ we want to learn a function $f : \mathcal{U} \times \mathcal{V} \to \mathbb{R}$ to make a prediction. For example, for a given protein and a given ligand, one wants to predict the binding affinity based on a set of examples. Pairwise prediction models are fitted using a set of labeled examples: $S = \{(u_h, v_h, y_h) \mid h = 1, \ldots, n\}$ is a set of labeled dyads. Let $U = \{u_i \mid i = 1, \ldots, m\}$ and $V = \{v_j \mid j = 1, \ldots, q\}$ denote, respectively, the sets of distinct objects of both types in the training set with $m = |U|$ and $q = |V|$. If a dataset contains exactly one labeled instance for every dyad $(u, v) \in U \times V$, this dataset is denoted as being complete. For such datasets, the labels can be structured as a matrix $\mathbf{Y} \in \mathbb{R}^{m \times q}$, so that its rows are indexed by the objects in U and the columns by the objects in V.

In the setting considered here, we assume to possess two positive definite kernel matrices, describing the objects of U and V respectively. Using Kronecker kernel ridge regression (Waegeman et al., 2012), a model can be fitted to make predictions for new dyads. Suppose we want to make predictions only for in-sample objects $(u, v) \in U \times V$, we can directly obtain the matrix of predictions $\mathbf{F} \in \mathbb{R}^{m \times q}$ by solving

$$\min_{\mathbf{F}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{q} (\mathbf{F}_{ij} - \mathbf{Y}_{ij})^2 + \frac{C}{2} \mathrm{vec}(\mathbf{F})^\top [\mathbf{G} \otimes \mathbf{K}]^{-1} \mathrm{vec}(\mathbf{F}) \,, \tag{1}$$

with $\mathbf{K}$ (resp. $\mathbf{G}$) the Gram matrices for the objects U (resp. V) and C a regularization parameter which can be selected by cross-validation. This optimization problem has two parts. The first term is contains the loss function and ensures that the predictions are close to the observed labels. The second term is a regularization term that ensures that similar pairs have a similar label. See (Johnson & Zhang, 2008; Liu & Yang, 2015) for a more in-depth discussion. This is an example of transductive learning, as the pairs for which we want to make a prediction are known before fitting the model (Chapelle et al., 2006). The solution of the above minimization problem is given by

$$\begin{aligned} \mathrm{vec}(\mathbf{F}) &= [\mathbb{I} + C[\mathbf{G} \otimes \mathbf{K}]^{-1}]^{-1} \mathrm{vec}(\mathbf{Y}) \\ &= \mathbf{H} \mathrm{vec}(\mathbf{Y}) \,. \end{aligned}$$

Here, $\mathbf{H}$ is typically denoted as the hat matrix, linking the labels to the predictions. Computing the hat matrix explicitly is often prohibitory large, even for modest m and q. Starting from the eigenvalue decompositions of $\mathbf{K}$ and $\mathbf{G}$, one can compute $\mathbf{F}$ without needing the hat matrix in an intermediate step. Using some algebraic manipulations, $\mathbf{F}$ can be obtained with a time complexity of $\mathcal{O}(m^3 + q^3)$ and space requirement of $\mathcal{O}(m^2 + q^2)$, provided that the dataset is complete. For example, for a protein-ligand interaction dataset of hundreds of proteins and thousands of ligands, the complexity of solving (1) is dominated by the numbers of objects (thousands of proteins) rather than the number of labels (hundreds of thousands pairwise interaction values).

Often, the training data is not complete. Hence, why

Preliminary work. Under review for Benelearn 2017. Do not distribute.

one needs a model to impute these missing values in the first place. For incomplete data, we need to solve a modification of (1):

$$\min_{\mathbf{F}} \frac{1}{2} \sum_{(i,j)\in\mathcal{T}} (\mathbf{F}_{ij} - \mathbf{Y}_{ij})^2 + \frac{C}{2} \mathrm{vec}(\mathbf{F})^\top [\mathbf{G} \otimes \mathbf{K}]^{-1} \mathrm{vec}(\mathbf{F}) \,. \quad (2)$$

Solving the above problem is determined by the number of pairwise observations n and might be huge even for modest m and q. Ironically, having less data makes it much harder to compute $\mathbf{F}$ compared to the complete case.

Since computing $\mathbf{F}$ can be done efficiently when the dataset is complete, we suggest the following simple recipe to update the missing values:

1. Initialize the missing values of the unlabelled dyads, making the label matrix complete. This can be done by using the average of the observed labels or initalizing them to zero.
2. Fit a model using this label matrix. This step has a very low computational cost if the eigenvalue decompositions of $\mathbf{K}$ and $\mathbf{G}$ were already computed.
3. Update the missing values using the model.
4. Repeat steps 2 and 3 until convergence.

Formally, we can show that the above steps always converge to the unique minimizer of (1) and the error w.r.t. $\mathbf{F}$ decays as a geometric series.

2. Illustration: inpainting an image

The pairwise methods can also be applied to images, naturally represented as a matrix. Using suitable kernels, the Kronecker-based methods can be used as a linear image filter - see (Gonzalez & Woods, 2007) for an introduction. A black-and-white image is merely a matrix with intensity values for each pixel. Here, the only features for the rows and columns are the x- and y-coordinates of the pixels. For the rows (resp. columns) a kernel can be constructed that quantifies the distance between pixels in the vertical (resp. horizontal) direction. In the experiments, we use a standard radial basis kernel on the pixel coordinates for the rows and columns plugged in Kronecker kernel ridge regression with a regularization parameter $\lambda = 0.1$. We will illustrate the imputation algorithm on a benchmark image of a cup of coffee.

Figure 1 shows the example image of which parts were removed. We either randomly removed 1%, 10%, 50%, 90% or 99% of the pixels or removed a 100×400 pixels block from the image. Subsequently, the iterative imputation algorithm was used to impute the missing part of the image. The missing pixels were initialized with the average value of the remaining pixels in the image. The bottom of Figure 1 shows the mean squared error of the values of the imputed pixels as a function of the number of iterations of the algorithm. For reference purposes, the variance is also indicated, corresponding to the expected mean squared error of using the mean as imputation. In all cases, the algorithm could restore the image substantially better than the baseline. If the image is relatively complete, the imputation is quite fast; all imputations could be done in under a minute on a standard laptop. Figure 2 shows some of the image restorations. With 10% of the pixels missing, the restoration is visually indistinguishable of the original. Using only 10% of the pixels, a blurry image of the original can be produced. In the case where a block of the image is missing, a 'shadow' of the coffee cup can be seen, showing that the model can at least detect some high-level features of the image.

3. Conclusions

We presented a simple algorithm to impute missing values in a transductive pairwise learning setting. It can be shown that the algorithm always rapidly converges to the correct solution. This algorithm was illustrated on an example of inpainting an image. Given the importance of pairwise learning in domains such as molecular network inference (Vert, 2008; Schrynemackers et al., 2013), recommender systems (Lü et al., 2012) and species interactions prediction (Poisot et al., 2016; et al., 2017), we believe this algorithm to be a useful tool in a variety of settings.

References

Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-Supervised Learning*. MIT Press.

Gonzalez, R. C., & Woods, R. E. (2007). *Digital Image Processing*. Pearson.

Johnson, R., & Zhang, T. (2008). Graph-based semi-supervised learning and spectral kernel design. *IEEE Transactions on Information Theory, 54*, 275–288.

Liu, H., & Yang, Y. (2015). Bipartite edge prediction via transductive learning over product graphs.

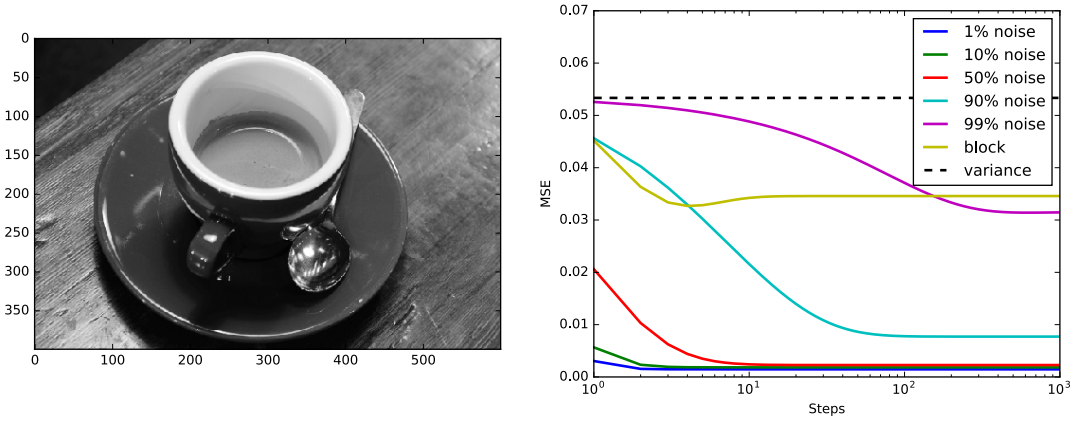


Figure 1. (left) An image of a cup of coffee. (right) Mean squared error of the imputed pixels as a function of the number of iterations of the imputation algorithm. Missing pixels are initiated with the average value of the observed pixels. The dotted line indicates the variance of the pixels, i.e. the approximate mean squared error of imputing with the average value of the imputed pixels.

Proceedings of the 32nd International Conference on Machine Learning (pp. 1880–1888).

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., & Zhou, T. (2012). Recommender systems. *Physics Reports*, *519*, 1–49.

, Poisot, T., Waegeman, W., & De Baets, B. (2017). Linear filtering reveals false negatives in species interaction data. *Scientific Reports*, *7*, 1–8.

Poisot, T., Stouffer, D. B., & Kéfi, S. (2016). Describe, understand and predict: why do we need networks in ecology? *Functional Ecology*, *30*, 1878–1882.

Schrynemackers, M., Küffner, R., & Geurts, P. (2013). On protocols and measures for the validation of supervised methods for the inference of biological networks. *Frontiers in Genetics*, *4*, 1–16.

Vert, J.-P. (2008). Reconstruction of biological networks by supervised machine learning approaches. In *Elements of computional systems biology*, 165–188.

Waegeman, W., Pahikkala, T., Airola, A., Salakoski, T., , & De Baets, B. (2012). A kernel-based framework for learning graded relations from data. *IEEE Transactions on Fuzzy Systems*, *20*, 1090–1101.

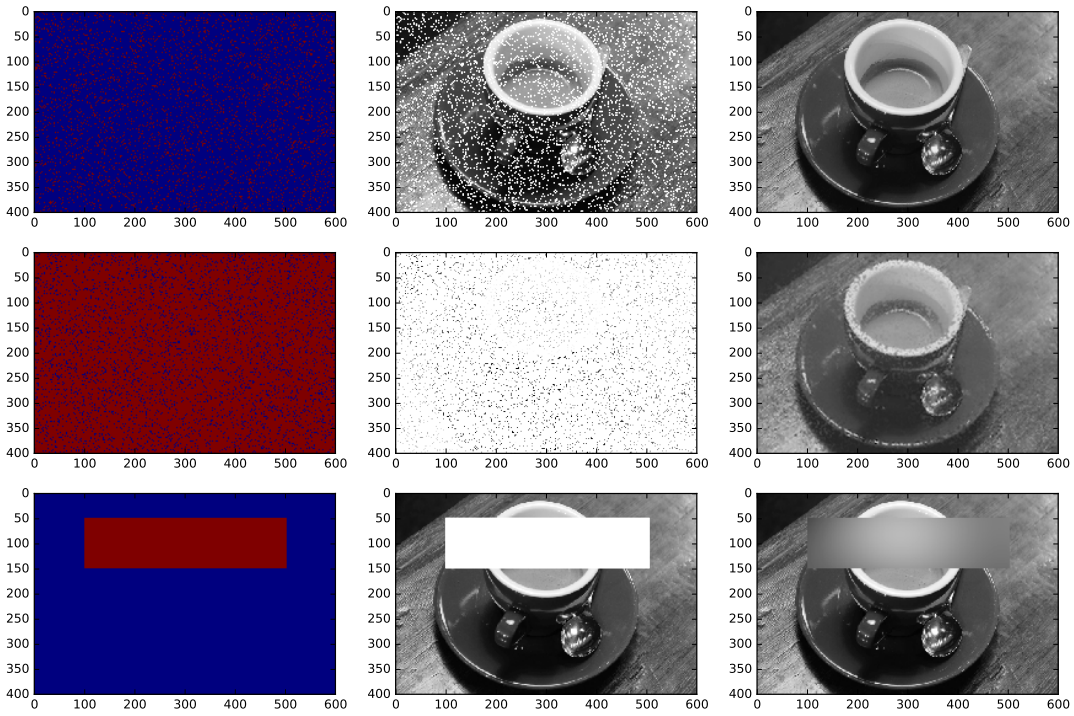


Figure 2. Examples of missing pixel imputation on the coffee image. (left) Mask indicating which pixels were removed, blue indicates available, red indicates missing. (middle) The coffee image with the corresponding missing pixels. (right) The restored image. (from top to bottom) 10% of the pixels randomly removed, 90% of the pixels randomly removed, a block of the image removed.

Towards an automated method based on Iterated Local Search optimization for tuning the parameters of Support Vector Machines

Sergio Consoli, Jacek Kustra, Pieter Vos, Monique Hendriks, Dimitrios Mavroeidis
[NAME.SURNAME]@PHILIPS.COM
Philips Research, High Tech Campus 34, 5656 AE Eindhoven, The Netherlands

Keywords: Support Vector Machines, Iterated Local Search, online tuning, parameters setting

Abstract

We provide preliminary details and formulation of an optimization strategy under current development that is able to automatically tune the parameters of a Support Vector Machine over new datasets. The optimization strategy is a heuristic based on Iterated Local Search, a modification of classic hill climbing which iterates calls to a local search routine.

1. Introduction

The performance of Support Vector Machine (SVM) strongly relies on the initial setting of the model parameters [Vapnik, 2000]. The parameters are usually set by training the SVM on a specific dataset and are then fixed when applied to a certain application. The automatic configuration for algorithms is faced with the same problem when doing hyper-parameter tuning in machine learning: finding the optimal setting of those parameters is an art by itself and as such much research on the topic has been explored in the literature [Ceylan & Taşkn, 2016, Lameski et al., 2015, Yang et al., 2012, Sherin & Supriya, 2015, Hutter et al., 2011]. Of the techniques used, grid search (or parameter sweep) is one of the most common methods to approximate optimal parameter values [Bergstra & Bengio, 2012]. Grid search involves an exhaustive search through a manually specified subset of the hyperparameter space of a learning algorithm, guided by some performance metric (e.g. cross-validation). This traditional approach, however, has several limitations: (i) it is vulnerable to local optimum; (ii) it does not provide any global optimality guarantee; (iii) setting an appropriate search interval is an ad-hoc approach; (iv) it is a computationally expensive approach, especially when search intervals require to capture wide ranges.

In this short contribution, we describe our preliminary investigation into an optimization method which tackles the parameter setting problem in SVMs using Iterated Local Search (ILS) [Lourenço et al., 2010], a popular explorative local search method used for solving discrete optimization problems. It belongs to the class of trajectory optimization methods, i.e. at each iteration of the algorithm the search process designs a trajectory in the search space, starting from an initial state and dynamically adding a new, better solution to the curve in each discrete time step. Iterated Local Search mainly consists of two steps. In the first step, a local optimum is reached by performing a walk in the search space, referred to as the *perturbation phase*. The second step is to efficiently escape from local optima by using an appropriate *local search phase* [Lourenço, 1995]. The application of an *acceptance criterion* to decide which of two local candidate solutions has to be chosen to continue the search process is also an important aspect of the algorithm. In the next section we provide the preliminary details and formulation of our optimization strategy based on ILS for the parameter tuning task in SVMs.

2. ILS for SVM parameters tuning

Given the input parameters $x \in X$ and their corresponding output parameters $y \in Y = \{-1, 1\}$, the separation between classes in SVMs is achieved by fitting the hyperplane $f(x)$ that has the optimal distance to the nearest data point used for training of any class: $f(x) = \sum_{i=1}^{n} \alpha_i y_i < x_i, x > +b$, where n is the total number of parameters. The goal in SVMs is to find the hyperplane which maximizes the minimum distances of the samples on each side of the

Preliminary work. Under review for Benelearn 2017. Do not distribute.

plane [Cortes & Vapnik, 1995]. A penalty is associated with the instances which are misclassified and added to the minimization function. This is done via the parameter C in the minimization formula: $\arg\min_{f(x)=\omega^T x+b} \frac{1}{2}\|\omega\|^2 + C\sum_i^n c(f, x_i, y_i)$.

By varying C, a trade-off between the accuracy and and stability of the function is defined. Larger values of C result in a smaller margin, leading to potentially more accurate classifications, however overfitting can occur. A mapping of the data with appropriate kernel functions $k(x, x')$ into a richer feature space, including non-linear features is applied prior to the hyperplane fitting. Among several kernels in the literature, we consider the Gaussian radial-basis function (RBF): $K(x_i, x') = exp(-\gamma\|x_i - x'\|^2), \gamma > 0$, where γ defines the variance of the RBF, practically defining the shape of the kernel function peaks: lower γ values set the bias to low and corresponding high γ to high bias.

The proposed ILS under current implementation for SVM tuning uses grid search [Bergstra & Bengio, 2012] as an inner local search routine, which is then iterated in order to make it fine-grained and finally producing the best parameters C and γ found to date. Given a training dataset D and an SVM model Θ, the procedure first generates an initial solution. We use an initial solution produced by grid search. The grid search exhaustively generates candidates from a grid of the parameter values, C and γ, specified in the arrays $range_\gamma \in \Re^+$ and $range_C \in \Re^+$. We choose arrays containing five different values for each parameter, so that the grid search method will look to 25 different parameters combinations. The range values are taken as different powers of 10 from -2 to 2. Solution quality is evaluated as the accuracy of the SVM by means of k-fold cross validation [McLachlan et al., 2004], and stored in the variable *Acc*.

Afterwards, the *perturbation phase*, which represents the core idea of ILS, is applied to the incumbent solution. The goal is to provide a good starting point (i.e. parameter ranges) for the next *local search phase* of ILS (i.e. the grid search in our case), based on the previous search experience of the algorithm, so as to obtain a better balance between exploration of the search space against wasting time in areas that are not giving good results. Ranges are set as: $range_\gamma = [\gamma * 10^{-2}, \gamma * 10^{-1}, \gamma, \gamma * 10, \gamma * 10^2] \equiv [\gamma_{inf-down}, \gamma_{inf-up}, \gamma, \gamma_{sup-down}, \gamma_{sup-up}]$, and $range_C = [C * 10^{-2}, C * 10^{-1}, C, C * 10, C * 10^2] \equiv [C_{inf-down}, C_{inf-up}, C, C_{sup-down}, C_{sup-up}]$.

Imagine that the grid search gets the set of parameters γ', C' as a new incumbent solution, whose evaluated accuracy is Acc'. Then the *acceptance criterion* of this new solution is that it produces a better quality, that is an increased accuracy, than the best solution to date. If it does not happen, the new incumbent solution is rejected and the ranges are updated automatically with the following values: $\gamma_{inf-down} = rand(\gamma_{inf-down} * 10^{-1}, \gamma_{inf-down})$ and $C_{inf-down} = rand(C_{inf-down} * 10^{-1}, C_{inf-down})$, $\gamma_{inf-up} = rand(\frac{\gamma-\gamma_{inf-up}}{2}, \gamma)$ and $C_{inf-up} = rand(\frac{C-C_{inf-up}}{2}, C)$, $\gamma_{sup-down} = rand(\frac{\gamma_{sup-down}-\gamma}{2}, \gamma)$ and $C_{sup-down} = rand(\frac{C_{sup-down}-C}{2}, C)$, and $\gamma_{sup-up} = rand(\gamma_{sup-up} * 10)$ and $C_{sup-up} = rand(C_{sup-up} * 10)$. That is, indifferently for γ and C, the values of the *inf-down* and *sup-up* components are random values always taken farther the current parameter (γ or C), in order to increase the diversification capability of the metaheuristic; while the values of the *inf-up* and *sup-down* components are random values always taken closer the current parameter, in order to increase the intensification strength around the current parameter. This perturbation setting should allow a good balance among the intensification and diversification factors.

Otherwise, if in the acceptance criterion the new incumbent solution, γ' and C', is better than the current one, γ and C, i.e. $Acc' > Acc$, then this new solution becomes the best solution to date ($\gamma \leftarrow \gamma'$, $C \leftarrow C'$), and $range_\gamma$ and $range_C$ are updated as usual. This procedure continues iteratively until the termination conditions imposed by the user are satisfied, producing at the end the best combination of γ and C as output.

3. Summary and outlook

We considered the parameter setting task in SVMs by an automated ILS heuristic, which looks to be a promising approach. We are aware that a more detailed description of the algorithm is deemed necessary, along with a thorough computational investigation. This is currently object of ongoing research, including a statistical analysis and comparison of the proposed algorithm against the standard grid search, in order to quantify and qualify the improvements obtained. Further research will explore the application of this strategy to other SVM kernels, considering also a variety of big, heterogenous datasets.

References

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*, 281–305.

Ceylan, O., & Taşkn, G. (2016). SVM parameter selection based on harmony search with an application to hyperspectral image classification. *24th Signal Processing and Communication Application Conference (SIU)* (pp. 657–660).

Cortes, C., & Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297.

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *Proceedings of the 5th Learning and Intelligent OptimizatioN Conference (LION 5)* (p. 507523). Rome, Italy.

Lameski, P., Zdravevski, E., Mingov, R., & Kulakov, A. (2015). Svm parameter tuning with grid search and its impact on reduction of model over-fitting. In *Rough sets, fuzzy sets, data mining, and granular computing*, vol. 9437 of *Lecture Notes in Computer Science*, 464–474. Heidelberg, Germany: Springer-Verlag.

Lourenço, H. R. (1995). Job-shop scheduling: computational study of local search and large-step optimization methods. *European Journal of Operational Research*, *83*, 347–364.

Lourenço, H. R., Martin, O. C., & Stützle, T. (2010). *Iterated local search: Framework and applications*, 363–397. Boston, MA: Springer US.

McLachlan, G. J., Do, K.-A., & Ambroise, C. (2004). *Analyzing microarray gene expression data*. New York: John Wiley & Sons.

Sherin, B. M., & Supriya, M. H. (2015). Selection and parameter optimization of svm kernel function for underwater target classification. *Underwater Technology (UT), 2015 IEEE* (pp. 1–5).

Vapnik, V. N. (2000). *The nature of statistical learning theory*. New York: Springer-Verlag.

Yang, C., Ding, L., & Liao, S. (2012). Parameter tuning via kernel matrix approximation for support vector machine. *Journal of Computers*, *7*.

Multi-step-ahead prediction of volatility proxies

Jacopo De Stefani JACOPO.DE.STEFANI@ULB.AC.BE
Gianluca Bontempi GIANLUCA.BONTEMPI@ULB.AC.BE
MLG, Departement dInformatique, Université Libre de Bruxelles, Boulevard du Triomphe CP212, 1050 Brussels, Belgium

Olivier Caelen[1] OLIVIER.CAELEN@WORLDLINE.COM
Dalila Hattab[2] DALILA.HATTAB@EQUENSWORLDLINE.COM
Worldline SA/NV, R&D, Bruxelles, Belgium[1]/Equens Worldline, R&D, Lille (Seclin), France[2]

Keywords : financial time series, volatility forecasting, multiple-step-ahead forecast

Though machine learning techniques have been often used for stock prices forecasting, few results are available for market fluctuation prediction. Nevertheless, volatility forecasting is an essential tool for any trader wishing to assess the risk of a financial investment. The main challenge of volatility forecasting is that, since this quantity is not directly observable, we cannot predict its actual value but we have to rely on some observers, known as volatility proxies (Poon & Granger, 2003) based either on intraday (Martens, 2002) or daily data. Once a proxy is chosen, the standard approach to volatility forecasting is the well-known GARCH-like model(Andersen & Bollerslev, 1998). In recent years several hybrid approaches are emerging (Kristjanpoller et al., 2014; Dash & Dash, 2016; Monfared & Enke, 2014) which combine GARCH with a non-linear computational approach. What is common to the state-of-the art is that volatility forecasting is addressed as an univariate and one-step-ahead autoregressive (AR) time series problem.

The purpose of our work is twofold. First, we aim to perform a statistical assessment of the relationships among the most used proxies in the volatility literature. Second, we explore a NARX (Nonlinear Autoregressive with eXogenous input) approach to estimate multiple steps of the output given the past output and input measurements, where the output and the input are two different proxies. In particular, our preliminary results show that the statistical dependencies between proxies can be used to improve the forecasting accuracy.

1. Background

Three main types of proxies are available in the literature : the proxy $\sigma^{SD,n}$, the family of proxies σ^i and σ^G. The first proxy corresponds to the natural definition of volatility (Poon & Granger, 2003), as a rolling standard deviation over a past time window of size n

$$\sigma_t^{SD,n} = \sqrt{\frac{1}{n-1}\sum_{i=0}^{n-1}(r_{t-i} - \bar{r_n})^2}$$

where $r_t = \ln\left(\frac{P_t^{(c)}}{P_{t-1}^{(c)}}\right)$ is the daily continuously compounded return , $\bar{r_n}$ is the average over $\{t, \cdots, t-n+1\}$ and $P_t^{(c)}$ are the closing prices. The family of proxies σ_t^i is analytically derived in Garman and Klass (1980). The proxy $\sigma_t^G = \sqrt{\omega + \sum_{j=1}^{p}\beta_j(\sigma_{t-j}^G)^2 + \sum_{i=1}^{q}\alpha_i\varepsilon_{t-i}^2}$ is the volatility estimation returned by a GARCH (1,1) (Hansen & Lunde, 2005) where $\varepsilon_{t-i} \sim \mathcal{N}(0,1)$ and the coefficients $\omega, \alpha_i, \beta_j$ are fitted according to the procedure in (Bollerslev, 1986).

2. The relationship between proxies

The fact that several proxies have been defined for the same latent variable raises the issues of their statistical association. For this reason we computed the proxies, discussed above, on the 40 time series of the French stock market index CAC40 in the period ranging from 05-01-2009 to 22-10-2014 (approximately 6 years). This corresponds to 1489 OHLC (Opening, High, Low, Closing) samples for each time series. Moreover, we obtained the continuously compounded return and the volume variable (representing the number of trades in given trading day).

Figure 1 shows the aggregated correlation (over all the 40 time series) between the proxies, obtained by meta-analysis (Field, 2001). The black rectangles indicate the results of an hierarchical clustering using

(Ward Jr, 1963) with k=3. As expected, we can observe a correlation clustering phenomenon between proxies belonging to the same family, i.e. σ_t^i and $\sigma_t^{SD,n}$. The presence of σ_t^0 in the $\sigma_t^{SD,n}$ cluster can be explained by the fact that the former represents a degenerate case of the latter when $n = 1$. Moreover, we find a correlation between the volume and the σ_t^i family.

3. NARX proxy forecasting

We focus here on the multi-step-ahead forecasting of the proxy σ^G by addressing the question whether a NARX approach can be beneficial in terms of accuracy. In particular we compare a univariate multi-step-ahead NAR model $\sigma^G_{t+h} = f(\sigma^G_t, \cdots, \sigma^G_{t-m}) + \omega$ with a multi-step-ahead NARX model $\sigma^G_{t+h} = f(\sigma^G_t, \cdots, \sigma^G_{t-m}, \sigma^X_t, \cdots, \sigma^X_{t-m}) + \omega$, for a specific embedding order $m = 5$ and for different estimators of the dependency f.

In particular we compare a naive model (average of the past values), a GARCH(1,1), and two machine learning approaches : a feedforward Artificial Neural Networks (single hidden layer, implemented with R `nnet`) and a k-Nearest Neighbors (automatic leave-one-out selection of the number of neighbors). Multi-step-ahead prediction is returned by a direct forecasting strategy (Taieb, 2014). The MASE results (Hyndman and Koehler (2006)) from 10 out-of-sample evaluations (Tashman (2000)) in Table 1 show that both machine learning methods outperform the benchmark methods (naive and GARCH) and that the ANN model can take advantage of the additional information provided by the exogenous proxy. The results in Table 2 confirm that such conclusion remains consistent when moving from a single stock time series in a given market to an index time series (S&P500).

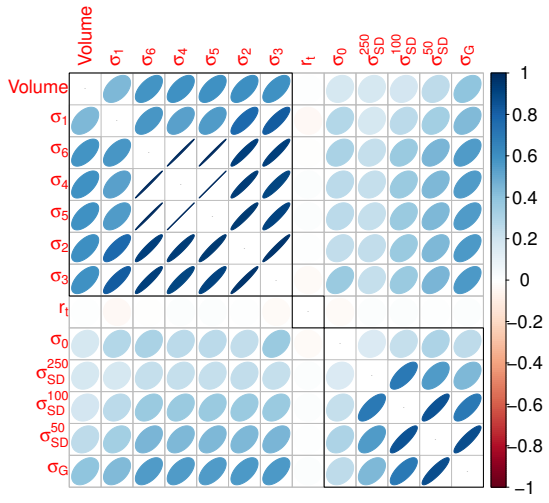


Figure 1. Summary of the correlations between different volatility proxies for the 40 CAC40 time series. Note that the continuously compounded return r_t has a very low correlation with all the other variables.

4. Conclusion and Future work

We studied the relationships between different proxies and we investigated the impact on the accuracy of volatility forecasting of three parameters : the choice of the exogenous proxy, the machine learning technique and the kind of autoregression. Results are preliminary for the moment. For the final version we expect to provide additional comparisons in terms of the number of series, forecasting horizons h model orders m.

Table 1. MASE (normalized wrt the accuracy of a naive method) for a 10-step volatility forecasting horizon on a single stock composing the CAC40 index on the period from 05-01-2009 to 22-10-2014, for different proxy combinations (rows) and different forecasting techniques (columns). The subscript X stands for the NARX model where σ^X is exogenous.

σ^X	ANN	kNN	ANN_X	kNN_X	GARCH(1,1)
σ^6	0.07	0.08	0.06	0.11	1.34
$Volume$	0.07	0.08	0.07	0.14	1.34
$\sigma^{SD,5}$	0.07	0.08	0.07	0.09	1.34
$\sigma^{SD,15}$	0.07	0.08	0.06	0.10	1.34
$\sigma^{SD,21}$	0.07	0.08	0.06	0.10	1.34

Table 2. MASE (normalized wrt the accuracy of a naive method) for a 10-step volatility forecasting horizon on the S&P500 index on the period from 01-04-2012 to 30-07-2013 as in the work of Dash & Dash, 2016, for different proxy combinations (rows) and different forecasting techniques (columns). The subscript X stands for the NARX model where σ^X is exogenous.

σ^X	ANN	kNN	ANN_X	kNN_X	GARCH(1,1)
σ^6	0.58	0.49	0.53	0.56	1.15
$Volume$	0.58	0.49	0.57	0.66	1.15
$\sigma^{SD,5}$	0.58	0.49	0.58	0.58	1.15
$\sigma^{SD,15}$	0.58	0.49	0.65	0.65	1.15
$\sigma^{SD,21}$	0.58	0.49	0.56	0.65	1.15

References

Andersen, T. G., & Bollerslev, T. (1998). Arch and garch models. *Encyclopedia of Statistical Sciences*.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, *31*, 307–327.

Dash, R., & Dash, P. (2016). An evolutionary hybrid fuzzy computationally efficient egarch model for volatility prediction. *Applied Soft Computing*, *45*, 40–60.

Field, A. P. (2001). Meta-analysis of correlation coefficients : a monte carlo comparison of fixed-and random-effects methods. *Psychological methods*, *6*, 161.

Garman, M. B., & Klass, M. J. (1980). On the estimation of security price volatilities from historical data. *Journal of business*, 67–78.

Hansen, P. R., & Lunde, A. (2005). A forecast comparison of volatility models : does anything beat a garch (1, 1) ? *Journal of applied econometrics*, *20*, 873–889.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, *22*, 679–688.

Kristjanpoller, W., Fadic, A., & Minutolo, M. C. (2014). Volatility forecast using hybrid neural network models. *Expert Systems with Applications*, *41*, 2437–2442.

Martens, M. (2002). Measuring and forecasting s&p 500 index-futures volatility using high-frequency data. *Journal of Futures Markets*, *22*, 497–518.

Monfared, S. A., & Enke, D. (2014). Volatility forecasting using a hybrid gjr-garch neural network model. *Procedia Computer Science*, *36*, 246–253.

Poon, S.-H., & Granger, C. W. (2003). Forecasting volatility in financial markets : A review. *Journal of economic literature*, *41*, 478–539.

Taieb, S. B. (2014). *Machine learning strategies for multi-step-ahead time series forecasting*. Doctoral dissertation, Ph. D. Thesis.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy : an analysis and review. *International journal of forecasting*, *16*, 437–450.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, *58*, 236–244.

Generalization Bound Minimization for Active Learning

Tom Viering T.J.Viering@tudelft.nl
TU Delft, Mekelweg 4, 2628 CD, Delft, The Netherlands

Jesse Krijthe JKrijthe@gmail.com
TU Delft, Mekelweg 4, 2628 CD, Delft, The Netherlands

Marco Loog M.Loog@tudelft.nl
TU Delft, Mekelweg 4, 2628 CD, Delft, The Netherlands

Keywords: active learning, learning theory, maximum mean discrepancy, generalization

Supervised machine learning models require enough labeled data to obtain good generalization performance. However, for many practical applications such as medical diagnosis or video classification it can be expensive or time consuming to label data (Settles, 2012). Often in practical settings unlabeled data is abundant, but due to high costs only a small fraction can be labeled. In active learning an algorithm chooses unlabeled samples for labeling (Cohn et al., 1994). The idea is that models can perform better with less labeled data if the labeled data is chosen carefully instead of randomly. This way active learning methods make the most of a small labeling budget or can be used to reduce labeling costs.

A generalization bound is an upper bound on the generalization error of the model that holds given certain assumptions. Several works have used generalization bounds to guide the active learning process (Gu & Han, 2012; Gu et al., 2012; Ganti & Gray, 2012; Gu et al., 2014). We have performed a theoretical and empirical study of active learners, that choose queries that explicitly minimize generalization bounds, to investigate the relation between bounds and their active learning performance. We limited our study to the kernel regularized least squares model (Rifkin et al., 2003) and the squared loss.

We studied the state-of-the-art Maximum Mean Discrepancy (MMD) active learner that minimizes a generalization bound (Chattopadhyay et al., 2012; Wang & Ye, 2013). The MMD is a divergence measure (Gretton et al., 2012) which is closely related to the Discrepancy measure (Mansour et al., 2009).

One of our novel theoretical results is a comparison of these bounds. We show that the Discrepancy bound on the generalization error is tighter than the MMD bound in the realizable setting — in this setting it is assumed there is no model mismatch. Tighter bounds are generally considered favorable as they estimate the generalization error more accurately. One might therefore also expect them to lead to better labeling choices in active learning when minimized and therefore we evaluated an active learner that minimizes the Discrepancy.

However, we observed that active learning using the tighter Discrepancy bound performs worse than the MMD. The underlying reason is that these bounds assume worst-case scenarios in order to derive their guarantees, and therefore minimizing these bounds for active learning may result in suboptimal performance in non-worst-case scenarios. In particular, the worst-case scenario assumed by the Discrepancy is, probabilistically speaking, very unlikely to occur compared to the scenario considered by the MMD and therefore the Discrepancy performs worse for active learning.

This insight lead us to introduce the Nuclear Discrepancy whose bound is looser. The Nuclear Discrepancy considers average case scenarios which occur more often in practice. Therefore, minimizing the Nuclear Discrepancy leads to an active learning strategy that is more suited to non-worst-case scenarios. Our experiments show that active learning using the Nuclear Discrepancy improves significantly upon the MMD and Discrepancy, especially in the realizable setting.

Our study illustrates that tighter bounds do not guarantee improved active learning performance and that a probabilistic analysis is essential: active learners should optimize their strategy for scenarios that are likely to occur in order to perform well in practice.

References

Chattopadhyay, R., Wang, Z., Fan, W., Davidson, I., Panchanathan, S., & Ye, J. (2012). Batch Mode Active Sampling Based on Marginal Probability Distribution Matching. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 741–749).

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, *15*, 201–221.

Ganti, R., & Gray, A. (2012). UPAL: Unbiased Pool Based Active Learning. *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)* (pp. 422–431).

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A Kernel Two-sample Test. *Machine Learning Research*, *13*, 723–773.

Gu, Q., & Han, J. (2012). Towards Active Learning on Graphs: An Error Bound Minimization Approach. *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM)* (pp. 882–887).

Gu, Q., Zhang, T., & Han, J. (2014). Batch-Mode Active Learning via Error Bound Minimization. *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI)*.

Gu, Q., Zhang, T., Han, J., & Ding, C. H. (2012). Selective Labeling via Error Bound Minimization. *Proceedings of the 25th Conference on Advances in Neural Information Processing Systems (NIPS)* (pp. 323–331).

Mansour, Y., Mohri, M., & Rostamizadeh, A. (2009). Domain Adaptation: Learning Bounds and Algorithms. *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*.

Rifkin, R., Yeo, G., & Poggio, T. (2003). Regularized least-squares classification. *Advances in Learning Theory: Methods, Model, and Applications*, *190*, 131–154.

Settles, B. (2012). Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *6*, 1–114.

Wang, Z., & Ye, J. (2013). Querying Discriminative and Representative Samples for Batch Mode Active Learning. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 158–166).

Projected Estimators for Robust Semi-supervised Classification

Jesse H. Krijthe JKRIJTHE@GMAIL.COM
Radboud University Nijmegen, Nijmegen, The Netherlands

Marco Loog M.LOOG@TUDELFT.NL
Delft University of Technology, Delft, The Netherlands

Keywords: Semi-supervised learning, Least Squares Classification, Projection

Abstract

For semi-supervised techniques to be applied safely in practice we at least want methods to outperform their supervised counterparts. We study this question for classification using the well-known quadratic surrogate loss function. Unlike other approaches to semi-supervised learning, the procedure proposed in this work does not rely on assumptions that are not intrinsic to the classifier at hand. Using a projection of the supervised estimate onto a set of constraints imposed by the unlabeled data, we find that it is possible to safely improve over the supervised solution in terms of this quadratic loss.

This abstract concerns the work presented in (Krijthe & Loog, 2017)

1. Problem & Setting

We consider the problem of semi-supervised classification using the quadratic loss function, which is also known as least squares classification or Fisher's linear discriminant classification (Hastie et al., 2009; Poggio & Smale, 2003). Suppose we are given an $N_l \times d$ matrix with feature vectors $\mathbf{X}$, labels $\mathbf{y} \in \{0,1\}^{N_l}$ and an $N_u \times d$ matrix with unlabeled objects $\mathbf{X}_\text{u}$ from the same distribution as the labeled objects. The goal of semi-supervised learning is to improve the classification decision function $f : \mathbb{R}^d \to \mathbb{R}$ using the unlabeled information in $\mathbf{X}_\text{u}$ as compared to the case where we do not have these unlabeled objects. In this work, we focus on linear classifiers where $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$.

Appearing in *Proceedings of Benelearn 2017*.

Much work has been done on semi-supervised classification, in particular on what additional assumptions about the unlabeled data may help improve classification performance. These additional assumptions, while successful in some settings, are less successful in others where they do not hold. In effect they can greatly deteriorate performance when compared to a supervised alternative (Cozman & Cohen, 2006). Since, in semi-supervised applications, the number of labeled objects may be small, the effect of these assumptions is often untestable. In this work, we introduce a conservative approach to training a semi-supervised version of the least squares classifier that is *guaranteed* to improve over the supervised least squares classifier, in terms of the quadratic loss measured on the labeled and unlabeled examples. To our knowledge this is the first approach that offers such strong, albeit conservative, guarantees for improvement over the supervised solution.

2. Sketch of the Approach

In the supervised setting, using a quadratic surrogate loss (Hastie et al., 2009), the following objective is minimized for $\mathbf{w}$:

$$L(\mathbf{w}, \mathbf{X}, \mathbf{y}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 . \tag{1}$$

The supervised solution $\mathbf{w}_\text{sup}$ is given by the minimization of (1) for $\mathbf{w}$. The well-known closed form solution to this problem is given by

$$\mathbf{w}_\text{sup} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} .$$

Our proposed semi-supervised approach is to project the supervised solution $\mathbf{w}_\text{sup}$ onto the set of all possible classifiers we would be able to get from some labeling of the unlabeled data.

$$\Theta = \left\{ \left(\mathbf{X}_\text{e}^\top \mathbf{X}_\text{e}\right)^{-1} \mathbf{X}_\text{e}^\top \begin{bmatrix} \mathbf{y} \\ \mathbf{y}_\text{u} \end{bmatrix} \mid \mathbf{y}_\text{u} \in [0,1]^{N_u} \right\} .$$

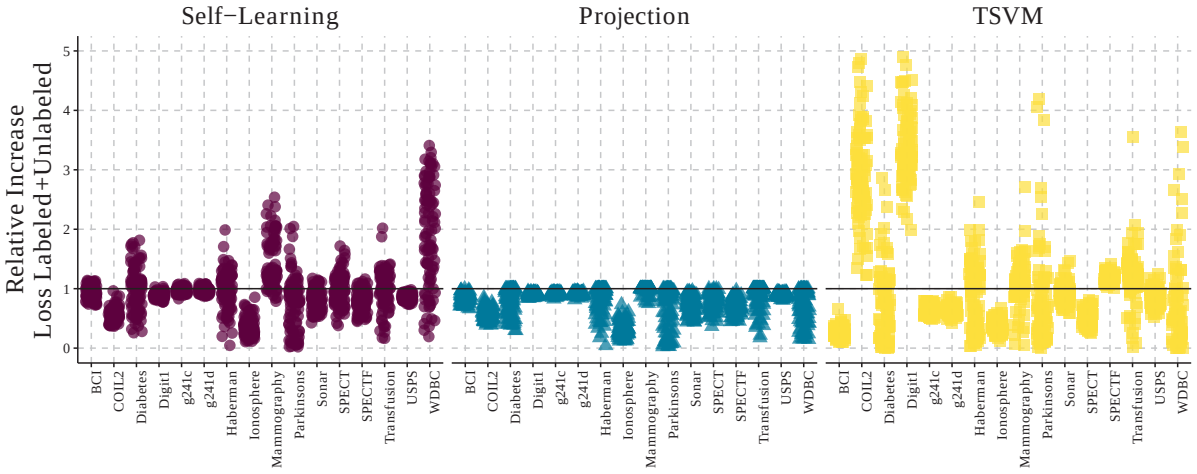


Figure 1. Ratio of the loss in terms of surrogate loss of semi-supervised and supervised solutions measured on the labeled and unlabeled instances. Values smaller than 1 indicate that the semi-supervised method gives a lower average surrogate loss than its supervised counterpart. Unlike the other semi-supervised procedures, the projection method, evaluated on labeled and unlabeled data, never has higher loss than the supervised procedure, as we prove in Theorem 1 of (Krijthe & Loog, 2017)

Note that this set, by construction, will also contain the solution $\mathbf{w}_{\text{oracle}}$, corresponding to the true but unknown labeling $\mathbf{y}_e^*$. Typically, $\mathbf{w}_{\text{oracle}}$ is a better solution than $\mathbf{w}_{\text{sup}}$ and so we would like to find a solution more similar to $\mathbf{w}_{\text{oracle}}$. This can be accomplished by projecting $\mathbf{w}_{\text{sup}}$ onto Θ.

$$\mathbf{w}_{\text{semi}} = \min_{\mathbf{w} \in \Theta} \mathrm{d}(\mathbf{w}, \mathbf{w}_{\text{sup}}),$$

where $\mathrm{d}(\mathbf{w}, \mathbf{w}')$ is a particular distance measure that measures the similarity between two classifiers. This is a quadratic programming problem with simple constraints that can be solved using, for instance, a simple gradient descent procedure.

3. Theoretical Guarantee

The main contribution of this work is a proof that the semi-supervised learner that we just described is guaranteed to never lead to worse performance than the supervised classifier, when performance is measured in terms of the quadratic loss on the labeled and unlabeled data. This property is shown empirically in Figure 1. This non-degradation property is important in practical applications, since one would like to be sure that the effort of the collection of, and computation with unlabeled data does not have an adverse effect. Our work is a conceptual step towards methods with these types of guarantees.

4. Empirical Evidence

Aside from the theoretical guarantee that performance never degrades when measured on the labeled and unlabeled training set in terms of the surrogate loss, experimental results indicate that it not only never degrades, but often improves performance. Our experiments also indicate the results hold when performance is evaluated on objects in a test set that were not used as unlabeled objects during training.

References

Cozman, F., & Cohen, I. (2006). Risks of Semi-Supervised Learning. In O. Chapelle, B. Schölkopf and A. Zien (Eds.), *Semi-supervised learning*, chapter 4, 56–72. MIT press.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The Elements of Statistical Learning*. Spinger. 2 edition.

Krijthe, J. H., & Loog, M. (2017). Projected Estimators for Robust Semi-supervised Classification. *Machine Learning*, http://arxiv.org/abs/1602.07865.

Poggio, T., & Smale, S. (2003). The Mathematics of Learning: Dealing with Data. *Notices of the AMS*, *50*, 537–544.

Towards an Ethical Recommendation Framework

Dimitris Paraschakis DIMITRIS.PARASCHAKIS@MAH.SE
Dept. of Computer Science, Malmö University, SE-205 06 Malmö, Sweden

Keywords: recommender systems, ethics of data mining / machine learning, ethical recommendation framework

Abstract

This study provides an overview of various ethical challenges that complicate the design of recommender systems (RS). The articulated ethical recommendation framework maps RS design stages to the corresponding ethical concerns, and further down to known solutions and the proposed user-adjustable controls. This framework aims to aid RS practitioners in staying ethically alert while taking morally charged design decisions. At the same time, it would give users the desired control over the sensitive moral aspects of recommendations via the proposed "ethical toolbox". The idea is embraced by the participants of our feasibility study.

1. Introduction

The notion of *recommendations* is built on real-life experiences and therefore perceived by humans as something inherently positive. User studies indeed show that the mere fact of labelling items as "recommendations" increases their chances of being consumed (Cremonesi et al., 2012; Knijnenburg, 2015). Whenever this fact is exploited for reasons beyond serving user needs, an ethical problem arises. Neglecting ethics in recommender systems may lead to privacy violation, identity theft, behavior manipulation, discrimination, offensive / hazardous content, misleading information, etc. Formally, we can define *recommendation ethics* as the study of the moral system of norms for serving recommendations of products and services to end users in the cyberspace. This system must account for moral implications stemming from both the *act* of recommending per se, and the enabling *technologies* involved. According to the recent study by (Tang & Winoto, 2016), there exist only three publications that specifically address the problem of ethical recommendations. Still, they only focus on particular problems in particular applications. A holistic view on the problem of recommendation ethics is currently lacking in the field, despite the massive research that RS attract nowadays. This problem is multifaceted and relates to several interconnected topics that we broadly group into the ethics of *data manipulation*, *algorithm design*, and *experimentation*. An in-depth discussion of these topics with accompanying examples is provided in our original work (Paraschakis, 2017). Here, we briefly outline the related issues in Section 2. In Section 3, we present an ethical recommendation framework that serves two purposes: a) it provides a roadmap for an ethics-inspired design of RS; b) it proposes a toolbox for manual tuning of morally-sensitive components of RS. We evaluate our proposal in Section 4 by presenting the results of the conducted survey. Section 5 concludes our work.

Preliminary work. Under review for Benelearn 2017. Do not distribute.

2. An Outline for Discussion

This section pinpoints RS-related ethical issues that are discussed in our original work (Paraschakis, 2017).

2.1. Ethics of data manipulation

- Informed consent for data collection/user profiling
- Data publishing: moral bonds between stakeholders, the failure of anonymization
- Sources of privacy breaches, possible attacks on RS users, GUI-based privacy solutions
- Content filtering and censorship

2.2. Ethics of algorithm design

- Algorithmic opacity, biases, and behavior manipulation
- Explanation interfaces and their challenges
- Price discrimination
- Filter bubbles via news recommendations

Table 1. General user-centric ethical recommendation framework

Design stage	Data collection	Data publishing	Algorithm design	User interface design	A/B testing
Ethical concerns	Privacy breaches, lack of awareness/consent, fake profile injection	Privacy / security / anonymity breaches	Biases, discrimination, behavior manipulation	Algorithmic opacity, content censorship	Fairness, side effects, lack of trust / awareness / consent
Known countermeasures	Informed consent, privacy-preserving collaborative filtering, identity verification	Privacy-preserving data publishing	Algorithm audits, reverse engineering, discrimination-aware data mining	Explanations, ethical rule set generation, content analysis	Informed consent, possibility to opt out and delete data
User-adjustable controls	"Do not track activity" tool	"Do not share data" tool	"Marketing bias" filter	"Content censorship" filter	"Opt out of experiments" tool
	This setting disallows the creation and maintenance of a user profile. Types of data can be manually defined and browsed items can be manually deleted (e.g. "manage history" tool on Amazon)	*This option allows local user profiling but forbids sharing data with third parties (even in the presence of anonymization). Types of data or categories of allowed recipients can be manually defined.*	*This filter is used to remove any business-driven bias introduced by RS providers, and set the recommendation engine to the "best match" mode (or other user-selectable modes, such as "cheapest first").*	*This tool can be used to set user-defined exclusion criteria for filtering out inappropriate items or categories. It also contains the option to turn the filter on and off (also with the possibility of scheduling).*	*This option can be used to reset the recommendation engine to its default algorithm, exclude the user from any future experiments, enable the opt-in option, delete data from previous experiments.*

2.3. Ethics of experimentation

- Famous cases of unethical A/B testing
- Three ways of consent acquisition for A/B testing
- Fairness and possibilities of user control

3. Summary as a Framework

Table 1 summarizes our findings in the form of a user-centric ethical recommendation framework, which maps RS design stages to potential ethical concerns and the recommended countermeasures. As a practical contribution, we propose an "ethical toolbox" comprised of user-adjustable controls corresponding to each design stage. These controls enable users to tune a RS to their individual moral standards. The usability of the provided controls may depend on many factors, such as their layout, frequency of using the system, sensitivity of data, and so on. As a vital first step, however, it is necessary to establish the general stance of users towards the ethics of recommender systems and whether the proposed toolbox would stand as a viable solution. This is done in the next section.

4. Feasibility study

We conduct an online survey[1] to find out people's opinions and their preferred course of action regarding five ethical issues of RS that are addressed by the proposed toolbox: *user profiling, data publishing, online experiments, marketing bias*, and *content censorship*. The survey was disseminated to Facebook groups of numerous European universities, yielding 214 responses from students and academic staff. The analysis of survey results immediately revealed participants' strong preference for taking morally sensitive issues under their control. In 4 out 5 studied issues, the majority voted for having a user-adjustable setting within a recommendation engine among other alternative solutions. The survey questions, responses, and analysis can be found in (Paraschakis, 2017).

[1] available at `http://recommendations.typeform.com/to/kgKNQO`

5. Conclusion

We conclude that multiple moral dilemmas emerge on every stage of RS design, while their solutions are not always evident or effective. In particular, there are many trade-offs to be resolved, such as user privacy vs. personalization, data anonymization vs. data utility, informed consent vs. experimentation bias, and algorithmic transparency vs. trade secrets. A careful risk assessment is crucial for deciding on the strategies of data anonymization or informed consent acquisition required for A/B testing or user profiling. We have found evidence that many big players on the RS market (Facebook, Amazon, Netflix, etc.) have faced loud ethics-related backlashes. Thus, it is important to ensure that a RS design is not only legally and algorithmically justified, but also *ethically* sound. The proposed framework suggests new paradigm of *ethics-awareness by design*, which utilizes existing technologies where possible, and complements them with user-adjustable controls. This idea was embraced by the vast majority of our survey participants, and future work should further test its usability in a fully implemented RS prototype.

References

Cremonesi, P., Garzotto, F., & Turrin, R. (2012). Investigating the persuasion potential of recommender systems from a quality perspective: An empirical study. *ACM Trans. Interact. Intell. Syst.*, *2*, 11:1–11:41.

Knijnenburg, B. (2015). *A user-tailored approach to privacy decision support*. Doctoral dissertation, University of California.

Paraschakis, D. (2017). Towards an ethical recommendation framework. *To appear in: Proceedings of the 11th IEEE International Conference on Research Challenges in Information Science*.

Tang, T., & Winoto, P. (2016). I should not recommend it to you even if you will like it: the ethics of recommender systems. *New Review of Hypermedia and Multimedia*, *22*, 111–138.

An Ensemble Recommender System for e-Commerce

Björn Brodén BJORN.BRODEN@APPTUS.COM
Apptus Technologies, Trollebergsvägen 5, SE-222 29 Lund, Sweden

Mikael Hammar MIKAEL.HAMMAR@APPTUS.COM
Apptus Technologies, Trollebergsvägen 5, SE-222 29 Lund, Sweden

Bengt J. Nilsson BENGT.NILSSON.TS@MAH.SE
Dept. of Computer Science, Malmö University, SE-205 06 Malmö, Sweden

Dimitris Paraschakis DIMITRIS.PARASCHAKIS@MAH.SE
Dept. of Computer Science, Malmö University, SE-205 06 Malmö, Sweden

Keywords: recommender systems, ensemble learning, thompson sampling, e-commerce, priming

Abstract

In our ongoing work we extend the Thompson Sampling (TS) bandit policy for orchestrating the collection of base recommendation algorithms for e-Commerce. We focus on the problem of item-to-item recommendations, for which multiple behavioral and content-based predictors are provided to an ensemble learner. The extended TS-based policy must be able to handle situations when bandit arms are non-stationary and non-answering. Furthermore, we investigate the effects of priming the sampler with pre-set parameters of reward distributions by analyzing the product catalog and/or event history, when such information is available. We report our preliminary results based on the analysis of two real-world e-Commerce datasets.

1. Introduction

A typical task in industrial e-Commerce applications is generating top-N item-to-item recommendations in a non-personalized fashion. Such recommendations are useful in "cold-start" situations when user profiles are very limited or non-existent, for example on landing pages. Even in this case, the cold-start problem manifests itself as a challenge of selecting items that are relevant to a given item. A natural way of tackling this issue is by following the "exploration-exploitation" paradigm of *multi-arm bandits* (MAB). These algorithms use reinforcement learning to optimize decision making in the face of uncertainty. Another established way of addressing the initial lack of data is to utilize *content-based filtering*, which recommends items based on their attributes. The efficacy of these two approaches along with the fact that many real-world recommenders tend to favor simple algorithmic approaches (Paraschakis et al., 2015), motivates the creation of an ensemble learning scheme consisting of a collection of base recommendation components that are orchestrated by a MAB policy. The proposed model has a number of advantages for a prospective vendor: a) it allows to easily plug-in/out recommendation components of any type, without making changes to the main algorithm; b) it is scalable because bandit arms represent algorithms and not single items; c) handling context can be shifted to the level of components, thus eliminating the need for contextual MAB policies. Our approach is detailed in the next section.

Preliminary work. Under review for Benelearn 2017. Do not distribute.

2. Approach

The modelling part can be split into two sub-problems:

1. Constructing base recommendation components
2. Choosing a bandit policy for the ensemble learner

2.1. Base recommendation components

We consider two types of components:

1. A **content-based** component defines the set of

items $\{y\}$ that share the same attribute value with the premise item x:
$x \longmapsto \{y : attribute_i(x) = attribute_i(y)\}$
For example, return all items of the same color.

2. A **collaborative filtering** component defines the set of items $\{y\}$ that are connected to the premise item x via a certain event type (click, purchase, addition to cart, etc.):
$x \longmapsto \{y : event_i(x)_t \rightarrow event_i(y)_{t'>t}\}$
For example, return all items that were bought after the premise item (across all sessions).

We note that special-purpose components can also be added by a vendor to handle all sorts of contexts.

2.2. Ensemble learner

The goal of our ensemble learner is to recommend top-N items for the premise item by querying the empirically best component(s). We employ the well-known Thompson Sampling (TS) policy (Chapelle & Li, 2011) for several practical reasons: a) its strong theoretical guarantees and excellent empirical performance; b) absence of parameters to tune; c) robustness to observation delays; d) flexibility in re-shaping arm reward distributions (see Section 2.3).
For a K-armed Bernoulli bandit, Thompson Sampling models the expected reward θ_a of each arm a as a Beta distribution with prior parameters α and β: $\theta_a \sim Beta(S_{a,t} + \alpha, F_{a,t} + \beta)$. In each round t, an arm with the highest sample is played. Success and failure counts $S_{a,t}$ and $F_{a,t}$ are updated according to the observed reward $r_{a,t}$.
The blind application of this classical TS model would fail in our case because of its two assumptions:

1. One arm pull per round. Because the selected component may return only few (or even zero!) items for a given query, pulling one arm at a time may not be sufficient to fill in the top-N recommendation list.
2. Arms are stationary. Because collaborative filtering components improve their performance over time, they have non-stationary rewards.

Therefore, our ongoing work extends Thompson Sampling to adapt to the task at hand. To address the first problem, we allow multiple arms to be pulled in each round and adjust the reward system accordingly. The second problem can be solved by dividing each component in sub-components of relatively stable behavior.

2.3. Priming the sampler

Apart from the proposed modifications of the TS model, we examine the effects of priming the sampler by pre-setting the prior parameters α and β of reward distributions. We consider two realistic scenarios where the estimation of these priors can be done:

1. Newly launched website. In this case, the estimation of the parameters relies solely on the analysis of the product catalog.
2. Pre-existing website. In this case, the estimation of the parameters can be done by utilizing the event history.

In both scenarios, we must be able to reason about the expected mean μ and variance σ^2 of reward distributions based on the analysis of the available data. We can then compute α and β as follows:

$$\alpha = -\frac{\mu\lambda}{\sigma^2} \qquad \beta = \frac{(\mu-1)\lambda}{\sigma^2}, \qquad \lambda = \sigma^2 + \mu^2 - \mu \quad (1)$$

3. Preliminary results and future work

In our preliminary experiments we compare TS to other popular bandit policies for the top-5 recommendation task, after making the adjustments proposed in Section 2.2. Two stand-alone recommenders are used as strong baselines: best sellers and co-purchases ("Those-Who-Bought-Also-Bought"). We run the experiments on two proprietary e-Commerce datasets of 1 million events each: a book store and a fashion store. The results below show the hit rate of each method. We observe that Thompson Sampling significantly outperforms the baselines and consistently outperforms state-of-the-art MAB policies by a small margin, which justifies our choice of method. Future work will demonstrate the predictive superiority of the extended TS in relation to the standard TS policy. Furthermore, we plan to examine what can be gained by priming the sampler and how exactly it can be done.

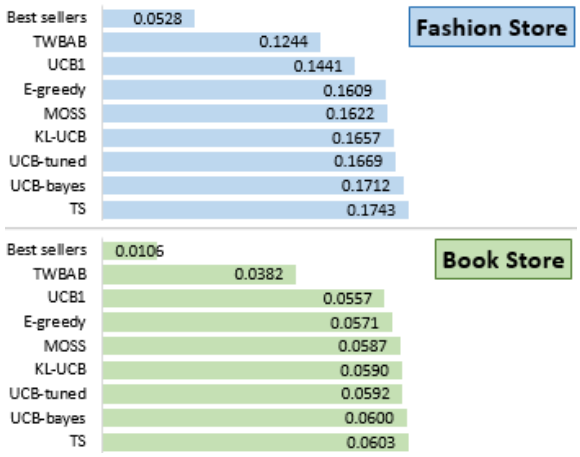


Figure 1. TS vs. baselines (measured in hit rate)

Acknowledgments

This research is part of the research projects "Automated System for Objectives Driven Merchandising", funded by the VINNOVA innovation agency; `http://www.vinnova.se/en/`, and "Improved Search and Recommendation for e-Commerce", funded by the Knowledge foundation; `http://www.kks.se`.

We express our gratitude to Apptus Technologies (`http://www.apptus.com`) for the provided datasets and computational resources.

References

Chapelle, O., & Li, L. (2011). An empirical evaluation of thompson sampling. *Proceedings of the 24th International Conference on Neural Information Processing Systems* (pp. 2249–2257).

Paraschakis, D., Holländer, J., & Nilsson, B. J. (2015). Comparative evaluation of top-n recommenders in e-commerce : an industrial perspective. *Proceedings of the 14th IEEE International Conference on Machine Learning and Applications* (pp. 1024–1031).

Ancestral Causal Inference (Extended Abstract)

Sara Magliacane SARA.MAGLIACANE@GMAIL.COM
VU Amsterdam, De Boelelaan 1083a, 1081 HV Amsterdam, The Netherlands

Tom Claassen T.CLAASSEN@CS.RU.NL
Radboud University Nijmegen, Postbus 9010, 6500GL Nijmegen, The Netherlands

Joris M. Mooij J.M.MOOIJ@UVA.NL
University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands

Keywords: causal inference, constraint-based causal discovery, structure learning

Abstract

This is an extended abstract of the NIPS 2016 paper (Magliacane et al., 2016).

Discovering causal relations from data is at the foundation of the scientific method. Traditionally, cause-effect relations have been recovered from experimental data in which the variable of interest is perturbed, but seminal work like the do-calculus (Pearl, 2009) and the PC/FCI algorithms (Spirtes et al., 2000; Zhang, 2008) demonstrate that, under certain assumptions, it is already possible to obtain significant causal information by using only observational data.

Recently, there have been several proposals for combining observational and experimental data to discover causal relations. These causal discovery methods are usually divided into two categories: constraint-based and score-based methods. Score-based methods typically evaluate models using a penalized likelihood score, while constraint-based methods use statistical independences to express constraints over possible causal models. The advantages of constraint-based over score-based methods are the ability to handle latent confounders naturally, no need for parametric modeling assumptions and an easy integration of complex background knowledge, especially for logic-based methods.

Two major disadvantages of constraint-based methods are: (i) vulnerability to errors in statistical independence test results, which are quite common in real-world applications, (ii) no ranking or estimation of the confidence in the causal predictions. Several approaches address the first issue and improve the reliability of constraint-based methods by exploiting redundancy in the independence information. Unfortunately, existing approaches have to choose to sacrifice either accuracy by using a greedy method (Claassen & Heskes, 2012; Triantafillou & Tsamardinos, 2015), or scalability by formulating a discrete optimization problem on a super-exponentially large search space (Hyttinen et al., 2014). Additionally, the second issue is addressed only in limited cases.

In (Magliacane et al., 2016) we propose Ancestral Causal Inference (ACI), a logic-based method that provides a comparable accuracy to the best state-of-the-art constraint-based methods as HEJ (Hyttinen et al., 2014), but improves on the scalability by using a more coarse-grained representation. Instead of representing direct causal relations, in ACI we represent and reason only with ancestral relations ("indirect" causal relations), which define an ancestral structure:

Definition 1. *An **ancestral structure** is any relation $\dashrightarrow$ on the observed variables that satisfies the non-strict partial order axioms:*

$$(\text{reflexivity}): X \dashrightarrow X,$$
$$(\text{transitivity}): X \dashrightarrow Y \ \wedge \ Y \dashrightarrow Z \implies X \dashrightarrow Z,$$
$$(\text{antisymmetry}): X \dashrightarrow Y \wedge Y \dashrightarrow X \implies X = Y.$$

Though still super-exponentially large, this representation drastically reduces computation time, as shown in the evaluation. Moreover, this representation turns out to be very convenient, because in real-world applications the distinction between direct causal relations and ancestral relations is not always clear or necessary.

To solve the vulnerability to statistical errors in inde-

Preliminary work. Under review for Benelearn 2017. Do not distribute.

pendence tests, ACI reformulates causal discovery as an optimization problem. Given a list I of weighted input statements (i_j, w_j), where i_j is the input statement (e.g. an independence test result) and w_j is the associated weight (representing its confidence), we define the loss function as the sum of the weights of the input statements that are not satisfied in $A \in \mathcal{A}$, where $\mathcal{A}$ is the set of all possible ancestral structures:

$$L(A, I) := \sum_{(i_j, w_j) \in I:\, i_j \text{ is not satisfied in } A} w_j.$$

We explore two simple weighting schemes:

- a *frequentist* approach, in which for any appropriate frequentist statistical test with independence as null hypothesis, we define the weight:

  $w = |\log p - \log \alpha|$, where $p = p$-value of the test

  and α is the significance level (e.g., 5%);

- a *Bayesian* approach, in which the weight of each input i using data set $\mathcal{D}$ is:

  $$w = \log \frac{p(i|\mathcal{D})}{p(\neg i|\mathcal{D})} = \log \frac{p(\mathcal{D}|i)}{p(\mathcal{D}|\neg i)} \frac{p(i)}{p(\neg i)},$$

  where the prior probability $p(i)$ can be used as a tuning parameter.

Under the standard assumptions for causal discovery (i.e. the Causal Markov and Faithfulness assumption), ACI implements five rules that relate ancestral relations and input independences, defining which inputs are not satisfied in a given ancestral structure.

For example, for X, Y, $\boldsymbol{W}$ disjoint (sets of) variables, one of the ACI rules is:

$$(X \perp\!\!\!\perp Y \mid \boldsymbol{W}) \wedge (X \not\dashrightarrow \boldsymbol{W}) \implies X \not\dashrightarrow Y,$$

where $X \perp\!\!\!\perp Y \mid \boldsymbol{W}$ represents the conditional independence of X and Y conditioning on $\boldsymbol{W}$, while $X \not\dashrightarrow Y$ represents the fact that X is not a cause of Y.

Using this loss function, we propose a method to score predictions according to their confidence. This is very important for practical applications, as the low reliability of the predictions of constraint-based methods has been a major impediment to their widespread usage. We define the confidence score for a statement s as:

$$C(s) = \min_{A \in \mathcal{A}} L(A,\ I + (\neg s, \infty)) - \min_{A \in \mathcal{A}} L(A,\ I + (s,\ \infty))$$

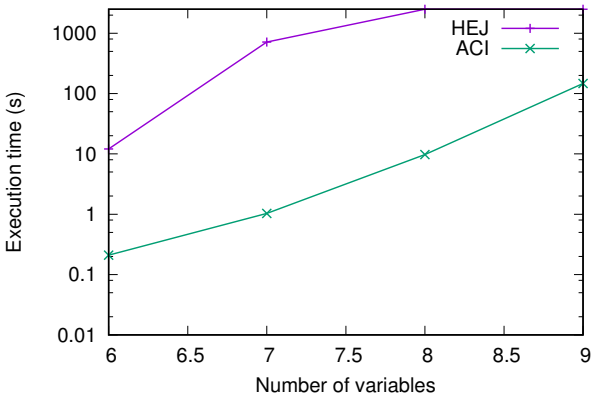


Figure 1. Synthetic data: execution times (log-scale).

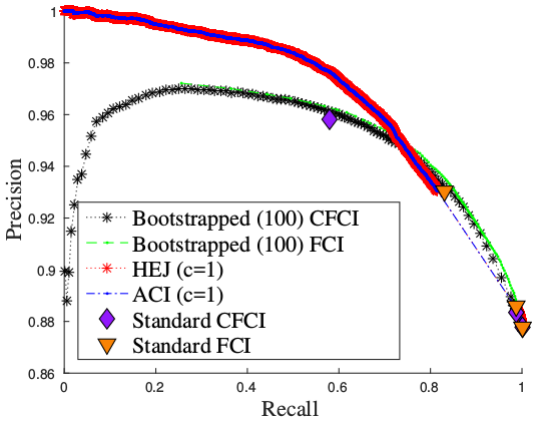


Figure 2. Synthetic data: example prediction-recall curve for non-causal predictions for 6 variables, maximum conditioning set $c = 1$, frequentist test with $\alpha = 0.05$.

It can be shown that this score is an approximation of the marginal probability of the statement and that it satisfies certain theoretical guarantees, like soundness and asymptotic consistency, given certain reasonable assumptions on the weights of all input statements.

We evaluate on synthetic data and show that ACI can outperform the state-of-the-art (HEJ, equipped with our scoring method), achieving a speedup of several orders of magnitude (as summarised in Figure 1), while still providing a comparable accuracy, as we show in an example precision and recall curve in Figure 2. In the full paper, we also illustrate its practical feasibility by applying it on a challenging protein data set that so far had only been addressed with score-based methods.

References

Claassen, T., & Heskes, T. (2012). A Bayesian approach to constraint-based causal inference. *UAI* (pp. 207–216).

Hyttinen, A., Eberhardt, F., & Järvisalo, M. (2014). Constraint-based causal discovery: Conflict resolution with Answer Set Programming. *UAI* (pp. 340–349).

Magliacane, S., Claassen, T., & Mooij, J. M. (2016). Ancestral causal inference. *NIPS*.

Pearl, J. (2009). *Causality: models, reasoning and inference*. Cambridge University Press.

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, prediction, and search*. MIT press.

Triantafillou, S., & Tsamardinos, I. (2015). Constraint-based causal discovery from multiple interventions over overlapping variable sets. *Journal of Machine Learning Research*, *16*, 2147–2205.

Zhang, J. (2008). On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artifical Intelligence*, *172*, 1873–1896.

Exceptional Model Mining in Ubiquitous and Social Environments

Martin Atzmueller M.ATZMULLER@UVT.NL
Tilburg University (TiCC), Warandelaan 2, 5037 AB Tilburg, The Netherlands

Keywords: exceptional model mining, subgroup discovery, community detection, social interaction networks

Abstract

Exceptional model mining in ubiquitous and social environments includes the analysis of resources created by humans (e. g., social media) as well as those generated by sensor devices in the context of (complex) interactions. This paper provides a structured overview on a line of work comprising a set of papers that focus on local exceptionality detection in ubiquitous and social environments and according complex social interaction networks.

1. Introduction

In ubiquitous and social environments, a variety of heterogenous multi-relational data is generated by sensors and social media (Atzmueller, 2012a). Then, a set of complex social interaction networks (Atzmueller, 2014), capturing distinct facets of the interaction space (Mitzlaff et al., 2014). Here, *local exceptionality detection* – based on *subgroup discovery* (Klösgen, 1996; Wrobel, 1997; Atzmueller, 2015) and *exceptional model mining* – provides flexible approaches for data exploration, assessment, and the detection of unexpected and interesting phenomena.

Subgroup discovery is an approach for discovering interesting subgroups – as an instance of *local pattern detection* (Morik, 2002). The interestingness is usually defined by a certain property of interest formalized by a quality function. In the simplest case, a binary target variable is considered, where the share in a subgroup can be compared to the share in the dataset in order to detect (exceptional) deviations. More complex target concepts consider sets of target variables. In particular, *exceptional model mining* (Leman et al., 2008; Duivesteijn et al., 2012; Duivesteijn et al., 2016) focuses on more complex quality functions.

Appearing in *Proceedings of Benelearn 2017*.

As a revision of (Atzmueller, 2016b), this paper summarizes formalizations and applications of subgroup discovery and exceptional model mining in the context of social interaction networks.

2. Methods

Social interaction networks (Atzmueller, 2014; Mitzlaff et al., 2011; Mitzlaff et al., 2013) focus on user-related social networks in social media capturing social relations inherent in social interactions, social activities and other social phenomena which act as proxies for social user-relatedness.

Exploratory data analysis is an important approach, e. g., for getting first insights into the data. In particular, descriptive data mining aims to uncover certain patterns for characterization and description of the data and the captured relations. Typically, the goal of description-oriented methods is not only to find an actionable model, but also a human interpretable set of patterns (Mannila, 2000).

Subgroup discovery and exceptional model mining are prominent methods for local exceptionality detection that can be configured and adapted to various analytical tasks. Local exceptionality detection especially supports the goal of explanation-aware data mining (Atzmueller & Roth-Berghofer, 2010), due to its more interpretable results, e. g., for characterizing a set of data, for concept description, for providing regularities and associations between elements in general, and for detecting and characterizing unexpected situations, e. g., events or episodes. In the following, we summarize approaches and methods for local exceptionality detection on attributed graphs, for behavioral characterization, and spatio-temporal analysis. Furthermore, we address issues of scalability and large-scale data processing.

2.1. Descriptive Community Detection

Communities can intuitively be defined as subsets of nodes of a graph with a dense structure in the corresponding subgraph. However, for mining such communities usually only structural aspects are taken into account. Typically, no concise nor easily interpretable community description is provided.

In (Atzmueller et al., 2016a), we focus on description-oriented community detection using subgroup discovery. For providing both structurally valid and interpretable communities we utilize the graph structure as well as additional descriptive features of the graph's nodes. We aim at identifying communities according to standard community quality measures, while providing characteristic descriptions at the same time. We propose several optimistic estimates of standard community quality functions to be used for efficient pruning of the search space in an exhaustive branch-and-bound algorithm. We present examples of an evaluation using five real-world data sets, obtained from three different social media applications, showing runtime improvements of several orders of magnitude. The results also indicate significant semantic structures compared to the baselines. A further application of this method to the exploratory analysis of social media using geo-references is demonstrated in (Atzmueller, 2014; Atzmueller & Lemmerich, 2013). Furthermore, a scalable implementation of the described description-oriented community detection approach is given in (Atzmueller et al., 2016b), which is also suited for large-scale data processing utilizing the Map/Reduce framework (Dean & Ghemawat, 2008).

2.2. Characterization of Social Behavior

Important structures that emerge in social interaction networks are given by subgroups. As outlined above, we can apply community detection in order to mine both the graph structure and descriptive features in order to obtain description-oriented communities. However, we can also analyze subgroups in a social interaction network from a compositional perspective, i. e., neglecting the graph structure. Then, we focus on the attributes of subsets of nodes or on derived parameters of these, e. g., corresponding to roles, centrality scores, etc. In addition, we can also consider sequential data, e. g., for characterization of exceptional link trails, i. e., sequential transitions, as presented in (Atzmueller, 2016a).

In (Atzmueller, 2012b), we discuss a number of exemplary analysis results of social behavior in mobile social networks, focusing on the characterization of links and roles. For that, we describe the configuration, adaptation and extension of the subgroup discovery methodology in that context. In addition, we can analyze multiplex networks by considering the match between different networks, and deviations between the networks, respectively. Outlining these examples, we demonstrate that local exceptionality detection is a flexible approach for compositional analysis in social interaction networks.

2.3. Exceptional Model Mining for Spatio-Temporal Analysis

Exploratory analysis on ubiquitous data needs to handle different heterogenous and complex data types. In (Atzmueller, 2014; Atzmueller et al., 2015), we present an adaptation of subgroup discovery using exceptional model mining formalizations on ubiquitous social interaction networks. Then, we can detect locally exceptional patterns, e. g., corresponding to bursts or special events in a dynamic network. Furthermore, we propose subgroup discovery and assessment approaches for obtaining interesting descriptive patterns and provide a novel graph-based analysis approach for assessing the relations between the obtained subgroup set. This exploratory visualization approaches allows for the comparison of subgroups according to their relations to other subgroups and to include further parameters, e. g., geo-spatial distribution indicators. We present and discuss analysis results utilizing a real-world ubiquitous social media dataset.

3. Conclusions and Outlook

Subgroup discovery and exceptional model mining provide powerful and comprehensive methods for knowledge discovery and exploratory analyis in the context of local exceptionality detection. In this paper, we presented according approaches and methods, specifically targeting social interaction networks, and showed how to implement local exceptionality detection on both a methodological and practical level.

Interesting future directions for local exceptionality detection in social contexts include extended postprocessing, presentation and assessment options, e. g., (Atzmueller et al., 2006; Atzmueller & Puppe, 2008; Atzmueller, 2015). In addition, extensions to predictive modeling, e. g., link prediction (Scholz et al., 2013; Atzmueller, 2014) are interesting options to explore. Furthermore, extending the analysis of sequential data, e. g., based on Markov chains as exceptional models (Atzmueller et al., 2016c; Atzmueller, 2016a; Atzmueller et al., 2017), as well as group and network dynamics (Atzmueller et al., 2014; Kibanov et al., 2014) are further interesting options for future work.

References

Atzmueller, M. (2012a). Mining Social Media. *Informatik Spektrum*, *35*, 132 – 135.

Atzmueller, M. (2012b). Mining Social Media: Key Players, Sentiments, and Communities. *WIREs Data Mining and Knowledge Discovery*, *2*, 411–419.

Atzmueller, M. (2014). Data Mining on Social Interaction Networks. *JDMDH*, *1*.

Atzmueller, M. (2015). Subgroup Discovery. *WIREs Data Mining and Knowledge Discovery*, *5*, 35–49.

Atzmueller, M. (2016a). Detecting Community Patterns Capturing Exceptional Link Trails. *IEEE/ACM ASONAM*. Boston, MA, USA: IEEE.

Atzmueller, M. (2016b). Local Exceptionality Detection on Social Interaction Networks. *ECML-PKDD 2016* (pp. 485–488). Springer.

Atzmueller, M., Baumeister, J., & Puppe, F. (2006). Introspective Subgroup Analysis for Interactive Knowledge Refinement. *AAAI FLAIRS* (pp. 402–407). AAAI Press.

Atzmueller, M., Doerfel, S., & Mitzlaff, F. (2016a). Description-Oriented Community Detection using Exhaustive Subgroup Discovery. *Information Sciences*, *329*, 965–984.

Atzmueller, M., Ernst, A., Krebs, F., Scholz, C., & Stumme, G. (2014). On the Evolution of Social Groups During Coffee Breaks. *WWW 2014 (Companion)*. New York, NY, USA: ACM Press.

Atzmueller, M., & Lemmerich, F. (2013). Exploratory Pattern Mining on Social Media using Geo-References and Social Tagging Information. *IJWS*, *2*.

Atzmueller, M., Mollenhauer, D., & Schmidt, A. (2016b). Big Data Analytics Using Local Exceptionality Detection. In *Enterprise Big Data Engineering, Analytics, and Management*. IGI Global.

Atzmueller, M., Mueller, J., & Becker, M. (2015). *Exploratory Subgroup Analytics on Ubiquitous Data*, vol. 8940 of *LNAI*. Heidelberg, Germany: Springer.

Atzmueller, M., & Puppe, F. (2008). A Case-Based Approach for Characterization and Analysis of Subgroup Patterns. *Applied Intelligence*, *28*, 210–221.

Atzmueller, M., & Roth-Berghofer, T. (2010). The Mining and Analysis Continuum of Explaining Uncovered. *AI-2010*. London, UK: SGAI.

Atzmueller, M., Schmidt, A., & Kibanov, M. (2016c). DASHTrails: An Approach for Modeling and Analysis of Distribution-Adapted Sequential Hypotheses and Trails. *WWW 2016 (Companion)*. ACM Press.

Atzmueller, M., Schmidt, A., Kloepper, B., & Arnu, D. (2017). HypGraphs: An Approach for Analysis and Assessment of Graph-Based and Sequential Hypotheses. *New Frontiers in Mining Complex Patterns*. Heidelberg, Germany: Springer.

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, *51*, 107–113.

Duivesteijn, W., Feelders, A., & Knobbe, A. J. (2012). Different Slopes for Different Folks: Mining for Exceptional Regression Models with Cook's Distance. *ICDM* (pp. 868–876). ACM Press, New York.

Duivesteijn, W., Feelders, A. J., & Knobbe, A. (2016). Exceptional Model Mining. *DMKD*, *30*, 47–98.

Kibanov, M., Atzmueller, M., Scholz, C., & Stumme, G. (2014). Temporal Evolution of Contacts and Communities in Networks of Face-to-Face Human Interactions. *Sci China Information Sciences*, *57*.

Klösgen, W. (1996). Explora: A Multipattern and Multistrategy Discovery Assistant. In *Advances in Knowledge Discovery and Data Mining*. AAAI.

Leman, D., Feelders, A., & Knobbe, A. (2008). Exceptional Model Mining. *PKDD* (pp. 1–16). Springer.

Mannila, H. (2000). Theoretical Frameworks for Data Mining. *SIGKDD Explor.*, *1*, 30–32.

Mitzlaff, F., Atzmueller, M., Benz, D., Hotho, A., & Stumme, G. (2011). *Community Assessment using Evidence Networks*, vol. 6904 of *LNAI*. Springer.

Mitzlaff, F., Atzmueller, M., Benz, D., Hotho, A., & Stumme, G. (2013). User-Relatedness and Community Structure in Social Interaction Networks. *CoRR/abs*, *1309.3888*.

Mitzlaff, F., Atzmueller, M., Hotho, A., & Stumme, G. (2014). The Social Distributional Hypothesis. *Journal of Social Network Analysis and Mining*, *4*.

Morik, K. (2002). *Detecting Interesting Instances*, vol. 2447 of *LNCS*, 13–23. Springer Berlin Heidelberg.

Scholz, C., Atzmueller, M., Barrat, A., Cattuto, C., & Stumme, G. (2013). New Insights and Methods For Predicting Face-To-Face Contacts. *ICWSM*. AAAI.

Wrobel, S. (1997). An Algorithm for Multi-Relational Discovery of Subgroups. *Proc. PKDD-97* (pp. 78–87). Heidelberg, Germany: Springer.

PRIMPing Boolean Matrix Factorization through Proximal Alternating Linearized Minimization

Sibylle Hess SIBYLLE.HESS@TU-DORTMUND.DE
Katharina Morik KATHARINA.MORIK @CS.UNI-DORTMUND.DE
Nico Piatkowski NICO.PIATKOWSKI@TU-DORTMUND.DE
TU Dortmund University, Computer Science 8, Otto-Hahn-Str. 12, Dortmund, Germany

Keywords: Tiling, Boolean Matrix Factorization, Minimum Description Length principle, Proximal Alternating Linearized Minimization, Nonconvex-Nonsmooth Minimization

Abstract

We propose a novel Boolean matrix factorization algorithm to solve the tiling problem, based on recent results from optimization theory. We demonstrate the superior robustness of the new approach in the presence of several kinds of noise and types of underlying structure. Experimental results on image data show that the new method identifies interpretable patterns which explain the data almost always better than the competing algorithms.

1. Introduction

In a large range of data mining tasks such as Market Basket Analysis, Text Mining, Collaborative Filtering or DNA Expression Analysis, we are interested in the exploration of data which is represented by a binary matrix. Here, we seek for sets of columns and rows whose intersecting positions frequently feature a one. This identifies, e.g., groups of users together with their shared preferences, genes that are often co-expressed among several tissue samples, or words that occur together in documents describing the same topic.

The identification of r such sets of columns and rows is formally stated by a factorization of rank r. Thereby, the $m \times n$ data matrix is approximated by the Boolean product of two matrices $Y \in \{0,1\}^{m \times r}$ and $X \in \{0,1\}^{n \times r}$ such that $D \approx YX^T$. Assuming that data matrices compound a structural component, which can be expressed by a suitable factorization, and a haphazard component, denoted as noise $N = D - YX^T$, the objective is difficult to delineate: where to draw the line between structure and noise? Are there natural limitations on the rank of the factorization? To what extend may the groups of columns and rows, identified by the rank-one factorizations $Y_{\cdot s}X_{\cdot s}^T$ overlap? Miettinen and Vreeken (2014) successfully apply the *Minimum Description Length* (MDL) principle to reduce these considerations into one objective: exploit just as many regularities as serves the compression of the data. Identifying regularities with column-row interrelations, the description length counterbalances the complexity of the model (derived interrelations) and the fit to the data, measured by the size of the encoded data using the model. Thereby, an automatic determination of the factorization rank r is enabled.

Since the Boolean factorization of a given rank, yielding the smallest approximation error, cannot be approximated within any factor in polynomial time (unless **NP=P**), state-of-the art algorithms rely on heuristics. Various greedy methods are proposed in order to filter the structure from the noise by minimization of a suitable description length (Miettinen & Vreeken, 2014; Lucchese et al., 2014; Karaev et al., 2015). However, the experiments indicate at large that the quality considerably varies depending on the distribution of noise and characteristics of the dataset (Miettinen & Vreeken, 2014; Karaev et al., 2015).

For real-world datasets, it is difficult (if not impossible) to estimate these aspects, to choose the appropriate algorithm or to assess its quality on the given dataset. Believing that the unsteady performance is due to a lack of theoretical foundation, we introduce the method *Primp* (Hess et al., 2017) to numerically optimize a real-valued approximation of the cost measure as known from the algorithms KRIMP (Siebes

Preliminary work. Under review for Benelearn 2017. Do not distribute.

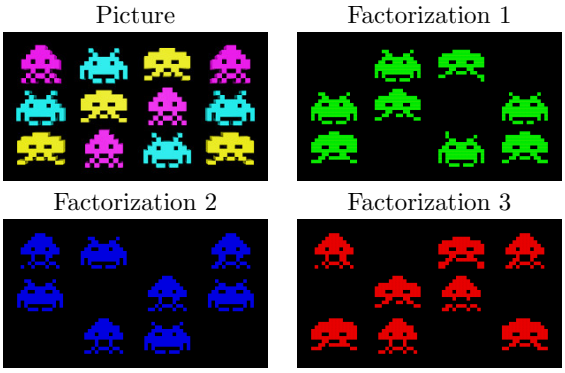


Figure 1. Reconstructions of the Space Invaders image on the top left by the three rank-one factorizations returned by PRIMP. Best viewed in color.

Algorithm 1 PRIMP(D, Δ_r, K)

1: $(X_K, Y_K) \leftarrow (\emptyset, \emptyset)$
2: **for** $r \in \{\Delta_r, 2\Delta_r, 3\Delta_r, \ldots\}$ **do**
3: $(X_0, Y_0) \leftarrow$INCREASERANK(X_K, Y_K, Δ_r)
4: **for** $k \in \{0, \ldots, K-1\}$ **do**
5: $\alpha_k^{-1} \leftarrow M_{\nabla_X F}(Y_k)$[1]
6: $X_{k+1} \leftarrow \mathrm{prox}_{\alpha_k \phi}\left(X_k - \alpha_k \nabla_X F(X_k, Y_k)\right)$
7: $\beta_k^{-1} \leftarrow M_{\nabla_Y F}(X_{k+1})$
8: $Y_{k+1} \leftarrow \mathrm{prox}_{\beta_k \phi}\left(Y_k - \beta_k \nabla_Y F(X_{k+1}, Y_k)\right)$
9: **end for**
10: $(X, Y) \leftarrow$ROUNDBINARY(X_K, Y_K)
11: **if** $r - r(X, Y) > 1$ **then**
12: **return** (X, Y)
13: **end if**
14: **end for**

et al., 2006) and SLIM (Smets & Vreeken, 2012). We assess the algorithms' ability to filter the *true* underlying structure from the noise. Therefore, we compare various performance measures in a controlled setting of synthetically generated data as well as for real-world data. We show that PRIMP is capable of recovering the latent structure in spite of varying database characteristics and noise distributions. In addition, we visualize the derived categorization into tiles by means of two image-datasets, demonstrating interpretability of the groupings as shown for one of the datasets in Fig. 1.

2. Primp

We sketch our method PRIMP, in Algorithm 1. A binary data matrix D, rank increment $\Delta_r \in \mathbb{N}$ and the maximum number of iterations K are the input of this algorithm. For every considered rank, PRIMP employs the *Proximal Alternating Linearized Minimization* (PALM) (Bolte et al., 2014) to numerically minimize the function

$$F(X, Y) + \phi(X) + \phi(Y),$$

where F is a smooth relaxation of the cost measure and ϕ penalizes non-binary values. Specifically, we choose $\phi(X) = \sum_{i,j} \Lambda(X_{ij})$, which employs the one-dimensional function

$$\Lambda(x) = \begin{cases} -|1 - 2x| + 1 & x \in [0, 1] \\ \infty & \text{otherwise.} \end{cases}$$

We show that the gradients $\nabla_X F$ and $\nabla_Y F$ are Lipschitz continuous with moduli $M_{\nabla_X F}$ and $M_{\nabla_Y F}$, which guarantees that the PALM, performed in lines 4-9, converges in a nonincreasing sequence of function values to a critical point.

The proximal mapping of ϕ, stated in line 6 and 8, is a function which returns a matrix satisfying the following minimization criterion:

$$\mathrm{prox}_\phi(X) \in \arg\min_{X^\star} \left\{ \frac{1}{2} \|X - X^\star\|^2 + \phi(X^\star) \right\}.$$

Loosely speaking, the proximal mapping gives its argument a little push into a direction which minimizes ϕ. We see in Algorithm 1 that the evaluation of this operator is a base operation. Thus, we derive a closed form of the proximal mapping of ϕ.

After the numerical minimization, the matrices X_K and Y_K, having entries between zero and one (ensured by the definition of ϕ), are rounded to binary matrices X and Y with respect to the minimization of the cost measure (line 10). If the rounding procedure returns binary matrices which use at least one (non-singleton) pattern less than possible ($r - r(X, Y) > 1$), the current factorization is returned. Otherwise, we increase the rank and add Δ_r random columns with entries between zero and one to the relaxed solution of the former iteration (X_K, Y_K) and numerically optimize again.

[1]Step sizes α_k and β_k have to be smaller than the inverse Lipschitz modulus in order to guarantee monotonic convergence. Thus, step sizes are multiplied with a constant smaller but close to one, in practice.

Acknowledgments

Part of the work on this paper has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Analysis", projects A1 and C1 `http://sfb876.tu-dortmund.de`.

References

Bolte, J., Sabach, S., & Teboulle, M. (2014). Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, *146*, 459–494.

Hess, S., Morik, K., & Piatkowski, N. (2017). The primping routine – tiling through proximal alternating linearized minimization (under minor revision). *Data Mining and Knowledge Discovery*.

Karaev, S., Miettinen, P., & Vreeken, J. (2015). Getting to know the unknown unknowns: Destructive-noise resistant boolean matrix factorization. *SDM* (pp. 325–333).

Lucchese, C., Orlando, S., & Perego, R. (2014). A unifying framework for mining approximate top-k binary patterns. *Transactions on Knowledge and Data Engineering*, *26*, 2900–2913.

Miettinen, P., & Vreeken, J. (2014). Mdl4bmf: Minimum description length for boolean matrix factorization. *ACM Trans. Knowl. Discov. Data*, *8*, 18:1–18:31.

Siebes, A., Vreeken, J., & van Leeuwen, M. (2006). Item sets that compress. *SDM* (pp. 393–404).

Smets, K., & Vreeken, J. (2012). Slim: Directly mining descriptive patterns. *SDM* (pp. 236–247). SIAM.

An expressive similarity measure for relational clustering using neighbourhood trees

Sebastijan Dumančić SEBASTIJAN.DUMANCIC@CS.KULEUVEN.BE
Department of Computer Science, KU Leuven, Belgium

Hendrik Blockeel HENDRIK.BLOCKEEL@CS.KULEUVEN.BE
Department of Computer Science, KU Leuven, Belgium

Keywords: Relational learning, Clustering, Similarity of structured objects

Abstract

In this paper, we introduce a novel similarity measure for relational data. It is the first measure to incorporate a wide variety of types of similarity, including similarity of attributes, similarity of relational context, and proximity in a hypergraph. We experimentally evaluate how using this similarity affects the quality of clustering on very different types of datasets. The experiments demonstrate that (a) using this similarity in standard clustering methods consistently gives good results, whereas other measures work well only on datasets that match their bias; and (b) on most datasets, the novel similarity outperforms even the best among the existing ones. This is a summary of the paper accepted to Machine Learning journal (Dumančić & Blockeel, 2017).

1. Introduction

In relational learning, the data set contains instances with relationships between them. Standard learning methods typically assume data are i.i.d. (drawn independently from the same population) and ignore the information in these relationships. Relational learning methods do exploit that information, and this often results in better performance. Much research in relational learning focuses on supervised learning (De Raedt, 2008) or probabilistic graphical models (Getoor & Taskar, 2007). Clustering, however, has received less attention in the relational context.

Preliminary work. Under review for Benelearn 2017. Do not distribute.

Clustering is an underspecified learning task: there is no universal criterion for what makes a good clustering, thus it is inherently subjective. This is known for i.i.d. data (Estivill-Castro, 2002), and even more true for relational data. Different methods for relational clustering have very different biases, which are often left implicit; for instance, some methods represent the relational information as a graph (which means they assume a single binary relation) and assume that similarity refers to proximity in the graph.

In this paper, we propose a very versatile framework for clustering relational data. It views a relational dataset as a hypergraph with typed vertices, typed hyperedges, and attributes associated to the vertices. The task we consider, is: cluster the vertices of one particular type. What distinguishes our approach from other approaches is that the concept of similarity used here is very broad. It can take into account attribute similarity, similarity of the relations an object participates in (including roles and multiplicity), similarity of the neighbourhood (in terms of attributes, relationships, or vertex identity), and interconnectivity or graph proximity of the objects being compared. We experimentally show that this framework for clustering is highly expressive and that this expressiveness is relevant, in the sense that on a number of relational datasets, the clusters identified by this approach coincide better with predefined classes than those of existing approaches.

2. Clustering over neighbourhood trees

2.1. Hypergraph Representation

Relational learning encompasses multiple paradigms. Among the most common ones are the *graph* view, where the relationships among instances are repre-

sented by a graph, and the *predicate logic* or equivalently *relational database* view, which typically assumes the data to be stored in multiple relations, or in a knowledge base with multiple predicates. Though these are in principle equally expressive, in practice the bias of learning systems differs strongly depending on which view they take. For instance, *shortest path distance* as a similarity measure is much more common in the graph view than in the relational database view. In the purely logical representation, however, no distinction is made between the constants that identify a domain object, and constants that represent the value of one of its features. Identifiers have no inherent meaning, as opposed to feature values.

In this work, we introduce a new view that combines elements of both. This view essentially starts out from the predicate logic view, but changes the representation to a hypergraph representation. Formally, the data structure that we assume in this paper is a typed, labelled hypergraph $H = (V, E, \tau, \lambda)$ with V being a set of vertices, and E a set of hyperedges; each hyperedge is an ordered set of vertices. The type function τ assigns a type to each vertex and hyperedge. A set of attributes $A(t)$ is associated with each $t \in T_V$. The labelling function λ assigns to each vertex a vector of values, one for each attribute of $A(\tau(v))$.

The clustering task we consider is the following: given a vertex type $t \in T_V$, partition the vertices of this type into clusters such that vertices in the same cluster tend to be similar, and vertices in different clusters dissimilar, for some subjective notion of similarity. In practice, it is of course not possible to use a subjective notion; one uses a well-defined similarity function, which hopefully in practice approximates well the subjective notion that the user has in mind. To be able to capture several interpretations of relational similarity, such as attribute or neighbourhood similarity, we represent each vertex with a *neighbourhood tree* - a structure that effectively describe a vertex and its neighbourhood.

2.2. Neighbourhood tree

Consider a vertex v. A neighbourhood tree aims to compactly represent the neighbourhood of the vertex v and all relationships it forms with other vertices, and it is defined as follows. For every hyperedge E in which v participates, add a directed edge from v to each vertex $v' \in E$. Label each vertex with its attribute vector. Label the edge with the hyperedge type and the position of v in the hyperedge (recall that hyperedges are ordered sets). The vertices thus added are said to be at depth 1. If there are multiple hyperedges connecting vertices v and v', v' is added each time it is encountered. Repeat this procedure for each v' on depth 1. The vertices thus added are at depth 2. Continue this procedure up to some predefined depth d. The root element is never added to the subsequent levels.

2.3. Similarity measure

The main idea behind the proposed dissimilarity measure is to express a wide range of similarity biases that can emerge in relational data, such as attribute or structural similarity. The proposed dissimilarity measure compares two vertices by comparing their neighbourhood trees. It does this by comparing, for each level of the tree, the distribution of vertices, attribute values, and outgoing edge labels observed on that level. Earlier work in relational learning has shown that distributions are a good way of summarizing neighbourhoods (Perlich & Provost, 2006).

The final similarity measure consists of a linear combination of different interpretations of similarity. Concretely, the similarity measure is a composition of components reflecting:

1. attributes of the root vertices,
2. attributes of the neighbouring vertices,
3. proximity of the vertices,
4. identity of the neighbouring vertices,
5. distribution of hyperedge types in a neighbourhood.

Each component is weighted by the corresponding weight w_i. These weights allow one to formulate an interpretation of the similarity between relational objects.

2.4. Results

We compared the proposed similarity measure against a wide range of existing relational clustering approaches and graph kernels on five datasets. The proposed similarity measure was used in conjunction with spectral and hierarchical clustering algorithms. We found that, on each separate dataset, our approach performs at least as well as the best competitor, and it is the only approach that achieves good results on all datasets. Furthermore, the results suggest that decoupling different sources of similarity into a linear combination helps to identify relevant information and reduce the effect of noise.

Acknowledgements

Research supported by Research Fund KU Leuven (GOA/13/010)

References

De Raedt, L. (2008). *Logical and relational learning.* Cognitive Technologies. Springer.

Dumančić, S., & Blockeel, H. (2017). An expressive dissimilarity measure for relational clustering using neighbourhood trees. *Machine Learning*, To Appear.

Estivill-Castro, V. (2002). Why so many clustering algorithms: A position paper. *SIGKDD Explor. Newsl.*, *4*, 65–75.

Getoor, L., & Taskar, B. (2007). *Introduction to statistical relational learning (adaptive computation and machine learning).* The MIT Press.

Perlich, C., & Provost, F. (2006). Distribution-based aggregation for relational learning with identifier attributes. *Mach. Learn.*, *62*, 65–105.

Complex Networks Track

Extended Abstracts

Dynamics Based Features for Graph Classification

Leonardo Gutiérrez Gómez LEONARDO.GUTIERREZ@UCLOUVAIN.BE
Jean-Charles Delvenne JEAN-CHARLES.DELVENNE@UCLOUVAIN.BE
Université catholique de Louvain, 4, Avenue Lemaître, B-1348 Louvain-la-Neuve, Belgium

Keywords: graph classification, dynamics on networks, machine learning on networks

Abstract

In this paper we propose a new feature based approach to network classification. We show how a dynamics on a network can be useful to reveal patterns about the organization of the components of the underlying graph where the process takes place. Measuring the auto-covariance along a random path on the network of a suitable set of network attributes including node labels, allows us to define generalized features across different time scales. These dynamic features turn out to be an appropriate discriminative signature of the network suitable for classification and recognition purposes. The method is tested empirically on established network benchmarks. Results show that our dynamic-based features are competitive and often outperform state of the art graph kernel based methods.

1. Introduction

A wide range of real world problems involve network analysis and prediction tasks. The complexity of social, engineering and biological networks make necessary developing methods to deal with the major difficulties mining graph-based data: the intrinsic high complexity of its structure and relations of its components and high dimensionality of the data.

In a typical graph classification task, we are interested in assign the most likely label to a graph among a set of classes. For example, in chemoinformatics, bioinformatics, one is interested in predicting the toxicity or anti-cancer activity of molecules. Characterization of proteins and enzymes is crucial in drugs research in order to discover the apparition of diseases. Social networks classification (Wang & Krim, 2012) is suitable for many social, marketing and targeting proposes, as well as mobility, collaboration networks and soon.

This problem has been treated before from the supervised and unsupervised machine learning perspective. In the first case, a set of discriminative hand-crafted features must be carefully selected in order to achieve high generalization capabilities. Typically it is done by manually choosing a set of structural, global and contextual features (Fei & Huan, 2008) from the underlying graph. Kernel-based methods (Hofmann et al., 2008) are very popular in this context. It consists in a two step process in which a suitable *kernel* function is devised capturing a similarity property of interest, followed by a classification step by using a kernelized version of a classification algorithm, i.e. logistic regression, support vector machines. Alternatively, unsupervised algorithms aims to learn those features from data, but at the cost of high training time and often blowing up the number of parameters, something clearly not suitable in the context of large social networks.

In that direction, understanding the structural decomposition of networks is crucial for our interest. Indeed, community detection or clustering algorithms on networks (Girvan & Newman, 2002) aim to disentangle meaningful, simplified patterns that are shared for groups of nodes along the network. In particular, dynamics-based approaches (Delvenne et al., 2013) as a general community detection framework, play a key role in our work. Certainly, when a dynamics takes place on a network, it is constrained by the network structure and then could potentially reveal interesting features of the organization of the network. Given this interdependence between dynamics and structure, we are able to extract meaningful features of the network across time scales, which will be useful for prediction purposes.

Preliminary work. Under review for Benelearn 2017. Do not distribute.

Dataset	GK	Deep GK	DF
COLLAB	72.84 ± 0.28	73.09 ± 0.25	**73.77 ± 0.22**
IMDB-BINARY	65.87 ± 0.98	66.96 ± 0.56	**70.32 ± 0.88**
IMDB-MULTI	43.89 ± 0.38	44.55 ± 0.52	**45.85 ± 1.18**
REDDIT-BINARY	77.34 ± 0.18	78.04 ± 0.39	**86.09 ± 0.53**
REDDIT-MULTI-5K	41.01 ± 0.17	41.27 ± 0.18	**51.44 ± 0.55**
REDDIT-MULTI-12K	31.82 ± 0.008	32.22 ± 0.10	**39.67 ± 0.42**

Table 1. Social networks: Mean and standard deviation of accuracy classification for Graphlet Kernel (GK) (Shervashidze et al., 2011), Deep Graphlet Kernel (Deep GK) (Yanardag & Vishwanathan, 2015), and Dynamic Features (DF, our method)

2. Method

In this work we explore the use of dynamics based graph features in the supervised graph classification setting. Having a well defined dynamic on a network, i.e a stationary random walk, we manually specify a candidate set of features based on our expertise and domain knowledge. That is, we chose a node feature such as degree, pagerank, local clustering coefficient etc, and look at the *autocovariance* (Delvenne et al., 2013) of this feature at times $t \in \{0, 1, 2, 3\}$ for a random walker jumping from node to node. This descriptors will be used as a global fingerprint of the network, describing generalized assortivities (the usual assortativity turns out to be the case $t = 1$), or clustering coefficient related for $t = 3$. In addition, for categorical node labels i.e. age, gender, atom type, we may use an association binary matrix H encoding node by class membership and then using the total autocovariance $H^T Cov(X_\tau, X_{\tau+t})H$, yielding even more features of interest.

3. Experiments and Results

This features are tested on many network benchmarks. For bioinformatics datasets (Figure 1) we run an automatic feature selection process by training an l_1 linear SVM classifier. On the other hand for social network datasets we opt for a Random Forest model (Table 1). We compare experimentally its classification accuracy with respect to a wide range of graph kernel methods. They are certainly the Graphlet kernel (Shervashidze et al., 2011), Shorthest path kernel (Borgwardt & Kriegel,) and the Weisfeiler-Lehman subtree kernel (Shervashidze et al., 2011), as well as their Deep kernel versions respectively (Yanardag & Vishwanathan, 2015). Random walks based kernels as p-step random walk (Smola & Kondor, 2003) and the random walk kernel (Gärtner et al., 2003) as well as the Ramon & Gartner kernel (Ramon & Gärtner, 2003) are also considered. Our results show that our method is capable to achieve and in many cases, outperform state of the art accuracies, in binary and multi-class graph classification tasks.

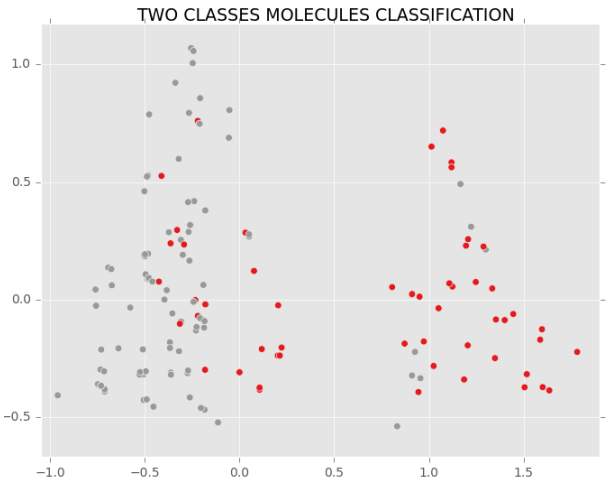


Figure 1. Dynamics based features are able to discriminate between molecules.

Acknowledgments

The authors acknowledges support from the grant "Actions de recherche concertées —Large Graphs and Networks" of the Communauté Française de Belgique. We also thank Marco Saerens and Roberto D'Ambrosio for helpful discussions and suggestions.

References

Barnett, I., Malik, N., Kuijjer, M. L., Mucha, P. J., & Onnela, J.-P. (2016). Feature-based classification of networks.

Borgwardt, K., & Kriegel, H. Shortest-path kernels on graphs. *Fifth IEEE International Conference on Data Mining (ICDM'05).*

Delvenne, J.-C., Schaub, M. T., Yaliraki, S. N., & Barahona, M. (2013). The stability of a graph partition: A dynamics-based framework for community detection. *Modeling and Simulation in Science, Engineering and Technology*, 221–242.

Fei, H., & Huan, J. (2008). Structure feature selection for graph classification. *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (pp. 991–1000). New York, NY, USA: ACM.

Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, *99*, 7821–7826.

Gärtner, T., Flach, P., & Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. *IN: CONFERENCE ON LEARNING THEORY* (pp. 129–143).

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, *36*, 1171–1220.

Ramon, J., & Gärtner, T. (2003). Expressivity versus efficiency of graph kernels. *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences* (pp. 65–74).

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., & Borgwardt, K. M. (2011). Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.*, *12*, 2539–2561.

Smola, A. J., & Kondor, R. (2003). *Kernels and regularization on graphs*, 144–158. Berlin, Heidelberg: Springer Berlin Heidelberg.

Wang, T., & Krim, H. (2012). Statistical classification of social networks. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Yanardag, P., & Vishwanathan, S. (2015). Deep graph kernels. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1365–1374). New York, NY, USA: ACM.

Improving Individual Predictions using Social Networks Assortativity

Dounia Mulders, Cyril de Bodt, Michel Verleysen {NAME.SURNAME}@UCLOUVAIN.BE
ICTEAM institute, Université catholique de Louvain, Place du Levant 3, 1348 Louvain-la-Neuve, Belgium

Johannes Bjelland JOHANNES.BJELLAND@TELENOR.COM
Telenor Research, Snarøyveien 30, N-1360 Fornebu, Norway

Alex (Sandy) Pentland PENTLAND@MIT.EDU
MIT Media Lab, Massachusetts Institute of Technology, 77 Mass. Ave., E14/E15, Cambridge, MA 02139, USA

Yves-Alexandre de Montjoye DEMONTJOYE@IMPERIAL.AC.UK
Data Science Institute, Imperial College London, 180 Queen's Gate, London SW7 2AZ, U.K.

Keywords: Belief propagation, assortativity, homophily, social networks, mobile phone metadata.

Abstract

Social networks are known to be assortative with respect to many attributes such as age, weight, wealth, ethnicity and gender. Independently of its origin, this assortativity gives us information about each node given its neighbors. It can thus be used to improve individual predictions in many situations, when data are missing or inaccurate. This work presents a general framework based on probabilistic graphical models to exploit social network structures for improving individual predictions of node attributes. We quantify the assortativity range leading to an accuracy gain. We also show how specific characteristics of the network can improve performances further. For instance, the gender assortativity in mobile phone data changes significantly according to some communication attributes.

1. Introduction

Social networks such as Facebook, Twitter, Google+ and mobile phone networks are nowadays largely studied for predicting and analyzing individual demographics (Traud et al., 2012; Al Zamal et al., 2012; Magno & Weber, 2014). Demographics are indeed a key input for the establishment of economic and social policies, health campaigns, market segmentation, etc. (Magno & Weber, 2014; Frias-Martinez et al., 2010; Sarraute et al., 2014). Especially in developing countries, such statistics are often scarce, as local censuses are costly, rough, time-consuming and hence rarely up-to-date (de Montjoye et al., 2014).

Social networks contain individual information about their users (e.g. generated tweets for Twitter), in addition to a graph topology information. The assortativity of social networks is defined as the nodes tendency to be linked to others which are similar in some sense (Aral et al., 2009). This assortativity with respect to various demographics of their individuals such as gender, age, weight, income level, etc. is well documented in the literature (McPherson et al., 2001; Madan et al., 2010; Wang et al., 2013; Smith et al., 2014; Newman, 2003). This property has been theorized to come either from influences or homophilies or a combination of both. Independently of its cause, this assortativity can be used for individual prediction purposes when some labels are missing or uncertain, e.g. for demographics prediction in large networks. Some methods are currently developed to exploit that assortativity (Al Zamal et al., 2012; Herrera-Yagüe & Zufiria, 2012). However, few studies take the global network structure into account (Sarraute et al., 2014; Dong et al., 2014). Also, to our best knowledge, no research quantifies how the performances are related to the assortativity strength. The goal of this work, already published, is to overcome these shortcomings (Mulders et al., 2017).

2. Method

We propose a framework based on probabilistic graphical models to exploit a social network structure, and

Appearing in *Proceedings of Benelearn 2017*.

especially its underlying assortativity, for individual prediction improvement in a general context. The network assortativity is quantified by the assortativity coefficient of the attribute to predict, denoted by r (Newman, 2003). The method can be applied with the only knowledge of the labels of a limited number of pairs of connected users in order to evaluate the assortativity, as well as class probability estimates for each user. These probabilities may for example be obtained by applying a machine learning algorithm exploiting the node-level information, after it has been trained on the individual data of the users with known labels. As described in (Mulders et al., 2017), a loopy belief propagation algorithm is applied on a Markov random field modeling the network to improve the accuracy of these prior class probability estimates. The model is able to benefit from the strength of the links, quantified for example by the number of contacts. The estimation of r allows to optimally tune the model parameters, by defining synthetic graphs. These simulations permit (1) to prevent overfitting a given network structure, (2) to perform the parameter tuning off-line and (3) to avoid requiring the labeled users to form a connected graph. These simulations also allow to quantify the assortativity range leading to an accuracy gain over an approach ignoring the network topology.

3. Mobile phone network

The methodology is validated on mobile phone data to predict gender (M and F resp. for male and female). Since our model exploits the gender homophilies, its performances depend on r. In the worst case of a randomly mixed network, $r = 0$. Perfect (dis-)assortativity leads to $r = (-)1$. In our network, $r \approx 0.3$, but Fig. 1 shows that it can change according to some communication attributes. The strongest edges (with many texts and/or calls) are more anti-homophilic, allowing to partition the edges into strong and weak parts, respectively disassortative and assortative ($r \approx 0.3$ in the weak part, whereas r can reach -0.3 in the strong one while retaining $\approx 1\%$ of the edges). This partition is exploited to improve the predictions by adapting the model parameters in the different parts of the network. Fig. 2 shows the accuracy and recall gains of our method, over simulated initial predictions with varying initial accuracies resulting from sampled class probability estimates. The highest accuracy gains are obtained in the range $[70, 85]\%$ of initial accuracy, covering the accuracies reached by state-of-the-art techniques aiming to predict gender using individual-level features (Felbo et al., 2015; Sarraute et al., 2014; Frias-Martinez et al., 2010). These gains overcome the results obtained with Sarraute et al's *reaction-diffusion* algorithm (2014).

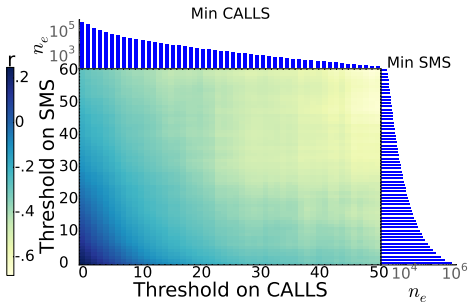


Figure 1. Gender assortativity coefficient in a mobile phone network when only the edges with a number of texts (SMS) and a number of calls (CALLS) larger than some increasing thresholds are preserved. The top (resp. right) histogram gives the number of edges (n_e) with CALLS (resp. SMS) larger than the corresponding value on the x-axis (resp. y-axis), on a log. scale.

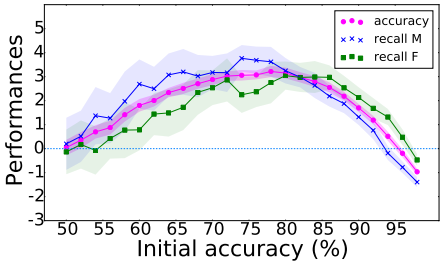


Figure 2. Accuracy and recall gains when varying the initial accuracy in a mobile phone network, averaged over 50 random simulations of the first predictions. The filled areas delimit intervals of one standard deviation around the mean gains.

4. Conclusion

This work shows how assortativity can be exploited to improve individual demographics prediction in social networks, using a probabilistic graphical model. The achieved performances are studied on simulated networks as a function of the assortativity and the quality of the initial predictions, both in terms of accuracy and distribution. Indeed, the relevance of the network information compared to individual features depends on (1) the assortativity amplitude and (2) the quality of the prior individual predictions. The graph simulations allow to tune the model parameters. Our method is validated on a mobile phone network and the model is refined to predict gender, exploiting both weak, homophilic and strong, anti-homophilic links.

Acknowledgments

DM and CdB are Research Fellows of the Fonds de la Recherche Scientifique - FNRS.

References

Al Zamal, F., Liu, W., & Ruths, D. (2012). Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. *ICWSM*, *270*.

Aral, S., Muchnik, L., & Sundararajan, A. (2009). Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, *106*, 21544–21549.

de Montjoye, Y.-A., Kendall, J., & Kerry, C. F. (2014). Enabling humanitarian use of mobile phone data. *Brookings Center for Tech. Innovation*.

Dong, Y., Yang, Y., Tang, J., Yang, Y., & Chawla, N. V. (2014). Inferring user demographics and social strategies in mobile social networks. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 15–24).

Felbo, B., Sundsøy, P., Pentland, A., Lehmann, S., & de Montjoye, Y.-A. (2015). Using deep learning to predict demographics from mobile phone metadata. *arXiv preprint arXiv:1511.06660*.

Frias-Martinez, V., Frias-Martinez, E., & Oliver, N. (2010). A gender-centric analysis of calling behavior in a developing economy using call detail records. *AAAI spring symposium: artificial intelligence for development*.

Herrera-Yagüe, C., & Zufiria, P. J. (2012). Prediction of telephone user attributes based on network neighborhood information. *International Workshop on Machine Learning and Data Mining in Pattern Recognition* (pp. 645–659).

Madan, A., Moturu, S. T., Lazer, D., & Pentland, A. S. (2010). Social sensing: obesity, unhealthy eating and exercise in face-to-face networks. *Wireless Health 2010* (pp. 104–110).

Magno, G., & Weber, I. (2014). International gender differences and gaps in online social networks. *International Conference on Social Informatics* (pp. 121–138).

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, 415–444.

Mulders, D., de Bodt, C., Bjelland, J., Pentland, A., Verleysen, M., & de Montjoye, Y.-A. (2017). Improving individual predictions using social networks assortativity. *Proceedings of the 12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM+)*.

Newman, M. E. (2003). Mixing patterns in networks. *Physical Review E*, *67*, 026126.

Sarraute, C., Blanc, P., & Burroni, J. (2014). A study of age and gender seen through mobile phone usage patterns in mexico. *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on* (pp. 836–843).

Smith, J. A., McPherson, M., & Smith-Lovin, L. (2014). Social distance in the united states: Sex, race, religion, age, and education homophily among confidants, 1985 to 2004. *American Sociological Review*, *79*, 432–456.

Traud, A. L., Mucha, P. J., & Porter, M. A. (2012). Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, *391*, 4165–4180.

Wang, Y., Zang, H., & Faloutsos, M. (2013). Inferring cellular user demographic information using homophily on call graphs. *INFOCOM, 2013 Proceedings IEEE* (pp. 3363–3368).

User-Driven Pattern Mining on knowledge graphs: an Archaeological Case Study

Wilcke, WX W.X.WILCKE@VU.NL
Department of Computer Science,
Department of Spatial Economics,
VU University Amsterdam, The Netherlands

de Boer, V V.DE.BOER@VU.NL
van Harmelen, FAH FRANK.VAN.HARMELEN@VU.NL
Department of Computer Science,
VU University Amsterdam, The Netherlands

Keywords: Knowledge Graph, Pattern Mining, Hybrid Evaluation, Digital Humanities, Archaeology

Abstract

In this work, we investigate to what extent data mining can contribute to the understanding of archaeological knowledge, published as knowledge graph, and which form would best meet the communities' needs. A case study was held which involved the user-driven mining of generalized association rules. Experiments have shown that the approach yielded mostly plausible patterns, some of which were rated as highly relevant by domain experts.

1. Introduction

Digital Humanities communities have recently began to show a growing interest in the *knowledge graph* as data modelling paradigm (Hallo et al., 2016). In this paradigm, knowledge is encoded as edges between vertices and is supported by semantic background knowledge. Already, many humanity data sets have been published as such, with large contributors being European archaeological projects such as CARARE and ARIADNE. These data have been made available in the *Linked Open Data* (LOD) cloud – an internationally distributed knowledge graph – bringing large amounts of structured data within arm's reach of archaeological researchers. This presents new opportunities for data mining (Rapti et al., 2015).

Appearing in *Proceedings of Benelearn 2017*.

In this work[1], we have investigated to what extent data mining can contribute to the understanding of archaeological knowledge, published as knowledge graph, and which form would best meet the communities' needs. For this purpose, we have constructed a pipeline which implements a state-of-the-art method to mine generalized association rules directly from the LOD cloud in an overall user-driven process (Freitas, 1999). Produced rules take the form: $\forall\chi(Type(\chi,t) \rightarrow (P(\chi,\phi) \rightarrow Q(\chi,\psi)))$. Their interestingness has been evaluated by a group of raters.

2. Approach

Our pipeline[2] facilitates the rule mining algorithm, various pre- and post-processing steps, and a simple rule browser. We will briefly touch on the most crucial components next:

Data Retrieval: On start, users are asked to provide a target pattern which defines their specific interest, e.g., ceramic artefacts. Optionally, users may specify numerous parameters which, if left empty, are set to defaults. Together, these are translated into a query which is used to construct an in-memory graph from the data retrieved from the LOD cloud.

Context Sampling: Entities that match the supplied target pattern (i.e., target entities) are ex-

[1]This research has been partially funded by the ARIADNE project through the European Commission under the Community's Seventh Framework Programme, contract no. FP7-INFRASTRUCTURES-2012-1-313193.

[2]Available at `github.com/wxwilcke/MINOS`.

tended with other entities related to them: their context. Unless specified by the user, contexts are sampled breath-first up to a depth of 3. This results in n subgraphs, with n equal to the total number of target entities in the in-memory graph. These subgraphs can be thought of as analogous to the instances in tabular data sets.

Pattern Mining: Our pipeline implements SWARM: a state-of-the-art generalized association rule mining algorithm (Barati et al., 2016). We motivate its selection by the algorithm's ability to exploit semantic background knowledge to generalize rules. In addition, the algorithm is transparent and yields interpretable results, thus fitting the domain requirements (Selhofer & Geser, 2014).

Dimension Reduction: A data-driven evaluation process is used to rate rules on their commonness. Hereto, we have extended the basic support and confidence measures with those tailored to graphs. Rules which are too rare or too common rules are omitted from the final result, as well as those with omnipresent relations (e.g., type and label). Remaining rules are shown in a simple faceted rule browser, which allows users to interactively customize templates (Klemettinen et al., 1994). For instance, to set acceptable ranges for confidence and support scores, as well as to specify the types of entities allowed in either or both antecedent and consequent.

3. Experiments

Experiments were run on an archaeological subset (±425k facts) of the LOD cloud[3], which contains detailed summaries about archaeological excavation projects in the Netherlands. Each summary holds information on 1) the project's organisational structure, 2) people and companies involved, 3) reports made and media created, 4) artefacts discovered together with their context and their (geospatial and stratigraphic) relation, and 5) fine-grained information about various locations and geometries.

Four distinct experiments have been conducted, each one having focussed on a different granularity of the data: A) project level, B) artefact level, C) context level, and D) subcontextual level. These were chosen together with domain experts, who were asked to describe the aspects of the data most interesting to them.

Results and Evaluation

Each experiment yielded more than 35,000 candidate rules. This has been brought down to several thousands using the aforementioned data-drive evaluation process. The remaining rules were then ordered on confidence (first) and support (second).

[3]Available at `pakbon-ld.spider.d2s.labs.vu.nl`.

Table 1. Normalized separate and averaged plausibility values (nominal scale) for experiments A through D as provided by three raters ($\kappa = -1.28e^{-3}$).

Experiment \ Rater	1	2	3	Mean
A	1.00	1.00	0.00	**0.67**
B	0.80	0.80	0.00	**0.53**
C	0.80	0.80	0.20	**0.60**
D	1.00	1.00	0.80	**0.93**
Mean	**0.90**	**0.90**	**0.25**	0.68

Table 2. Normalized separate and averaged relevancy values (ordinal scale) for experiments A through D as provided by three raters ($\kappa = 0.31$).

Experiment \ Rater	1	2	3	Mean
A	0.13±0.18	0.13±0.18	0.00±0.00	**0.09**±0.12
B	0.53±0.30	0.53±0.30	0.33±0.47	**0.47**±0.36
C	0.53±0.30	0.33±0.24	0.67±0.41	**0.51**±0.32
D	0.60±0.28	0.47±0.18	0.80±0.45	**0.62**±0.30
Mean	**0.45**±0.31	**0.37**±0.26	**0.45**±0.48	0.42±0.35

For each experiment, we selected 10 example rules from the top-50 candidates to create an evaluation set of 40 rules in total. Three domain experts were then asked to evaluate these on both plausibility and relevancy to the archaeological domain. Each rule was accompanied by a transcription in natural language to further improve its interpretability. For instance, a typical rule might state: *"For every artefact in the data set holds: if it consists of raw earthenware (Nimeguen), then it dates from early Roman to late Roman times"*.

The awarded plausibility scores (Table 1) indicate that roughly two-thirds of the rules (`0.68`) were rated plausible, with experiment D yielding the most by far. Rater 3 was far less positive than rater 1 and 2, and has a strong negative influence on the overall plausibility scores. In contrast, the relevancy scores (Table 2) are in fair agreement with an overall score of `0.42`, implying a slight irrelevancy. This can largely be attributed to experiment A, which scored considerably lower than the other experiments.

4. Conclusion

Our raters were positively surprised by the range of patterns that we were able to discover. Most of these were rated plausible, and some even as highly relevant. Nevertheless, trivialities and tautologies were also frequently encountered. Future research should focus on this by improving the data-driven evaluation step.

References

Barati, M., Bai, Q., & Liu, Q. (2016). *Swarm: An approach for mining semantic association rules from semantic web data*, 30–43. Cham: Springer International Publishing.

Freitas, A. A. (1999). On rule interestingness measures. *Knowledge-Based Systems*, *12*, 309–315.

Hallo, M., Luján-Mora, S., Maté, A., & Trujillo, J. (2016). Current state of linked data in digital libraries. *Journal of Information Science*, *42*, 117–127.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., & Verkamo, A. I. (1994). Finding interesting rules from large sets of discovered association rules. *Proceedings of the third international conference on Information and knowledge management* (pp. 401–407).

Rapti, A., Tsolis, D., Sioutas, S., & Tsakalidis, A. (2015). A survey: Mining linked cultural heritage data. *Proceedings of the 16th International Conference on Engineering Applications of Neural Networks (INNS)* (p. 24).

Selhofer, H., & Geser, G. (2014). *D2.1: First report on users needs* (Technical Report). ARIADNE. `http://ariadne-infrastructure.eu/Resources/D2.1-First-report-on-users-needs`.

Harvesting the right tweets: Social media analytics for the Horticulture Industry

Marijn ten Thij M.C.TEN.THIJ@VU.NL
Vrije Universiteit Amsterdam, Faculty of Sciences, Amsterdam, The Netherlands

Sandjai Bhulai S.BHULAI@VU.NL
Vrije Universiteit Amsterdam, Faculty of Sciences, Amsterdam, The Netherlands

Keywords: Twitter, horticulture, social media analytics

Abstract

In our current society, data has gone from scarce to superabundant: huge volumes of data are being generated every second. A big part of this flow is due to social media platforms, which provide a very volatile flow of information. Leveraging this information, which is buried in this fast stream of messages, poses a serious challenge. A vast amount of work is devoted to tackle this challenge in different business areas. In our work, we address this challenge for the horticulture sector, which has not received a lot of attention in the literature. Our aim is to extract information from the social data flow that can empower the horticulture sector. In this abstract, we present our first steps towards this goal.

In recent years, there have been a lot of overwhelming changes in how people communicate and interact with each other, mostly due to social media. It has revolutionized the Internet into a more personal and participatory medium. Consequently, social networking is now the top online activity on the Internet. With this many subscriptions to social media, massive amounts of information, accumulating as a result of interactions, discussions, social signals, and other engagements, form a valuable source of, which can be leveraged through social media analytics.

Social media analytics is the process of tracking conversations around specific phrases, words or brands [Fan & Gordon, 2014]. Through tracking, one can leverage these conversations to discover opportunities or to create content for those audiences. It requires advanced analytics that can detect patterns, track sentiment, and draw conclusions based on where and when conversations happen. Doing this is important for many business areas since actively listening to customers avoids missing out on the opportunity to collect valuable feedback to understand, react, and to provide value to customers.

The retail sector is probably the business area that utilizes social media analytics the most. More than 60% of marketeers use social media tools for campaign tracking [Järvinen et al., 2015], brand analysis [Hays et al., 2013], and for competitive intelligence [He et al., 2015][1]. Moreover, they also use tools for customer care, product launches, and influencer ranking. Social media analytics is also heavily used in news and journalism for building and engaging a news audience, and measuring those efforts through data collection and analysis [Bhattacharya & Ram, 2012, Castillo et al., 2014, Lehmann et al., 2013]. A similar use is also adopted in sports to actively engage with fans. In many business areas one also uses analytics for event detection [Lanagan & Smeaton, 2011] or automatic reporting of matches [Van Oorschot et al., 2012, Nichols et al., 2012] and user profiling [Xu et al., 2015].

In our work, we address this challenge for the horticulture sector, which has not received a lot of attention in the literature. The horticulture industry is a traditional sector in which producers are focused on production, and in which many traders use their own transactions as the main source of information. This leads to reactive management with very little anticipa-

Appearing in *Proceedings of Benelearn 2017*.

[1] `http://www.netbase.com/blog/why-use-social-analytics`

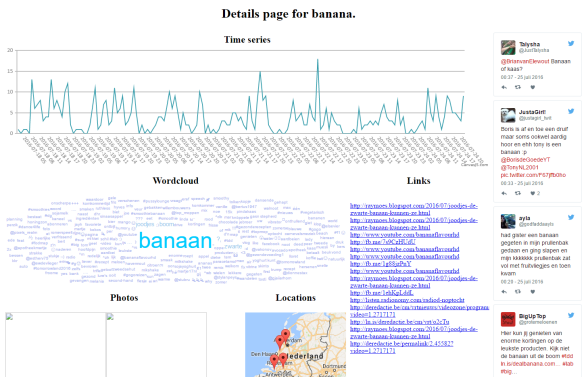


Figure 1. Example of details page for tweets mentioning bananas.

tion to events in the future. Growers and traders lack data about consumer trends and how the products are used and appreciated. This setting provides opportunities to enhance the market orientation of the horticulture industry, e.g., through the use of social media. Data on consumer's appreciation and applications of products are abundant on social media. Furthermore, grower's communication on social media might indicate future supply. This creates a need for analytic methods to analyze social media data and to interpret them. Here, we present our first steps towards this goal, as presented in [ten Thij et al., 2016].

The tweets that we use in this study are scraped using the filter stream of the Twitter Application Programming Interface (API)[2]. Since we do not have access to the full Twitter feed, we do not receive all tweets that we request due to rate limitations by Twitter[3]. Therefore, we use a list of 400 common tokens, that are frequently used in Dutch tweets. Then, we filter the tweets using two lists of product names, provided by our partners from GroentenFruitHuis and Floricode. One list contains fruits and vegetables, e.g., apple, orange, and mango, and the other contains flowers and plants, e.g., tulip, rose, and lily. After retrieving the tweets, the first step towards empowering the sector is knowing what kind of information can be retrieved from the social feed.

Using the data, we construct a weekly time series reflecting the number of mentions of the products. We compared these numbers to sales numbers of the most occurring product type of these products. Thus, we compared the number of Dutch tweets mentioning 'pears' or 'pear' to the number of Conference pears that are sold in the same time frame. Similarly, we compared the number of tweets mentioning 'apple' or 'apples' to the number of Elstar apples that are sold. We found that in both cases the time series for the tweets and the sales are comparable. Using an eight-hour shift for the sales time series, we find a Pearson correlation coefficient [Pearson, 1895] of 0.44 for the apples series and a coefficient of 0.46 for the pears series. These results indicate that it could be possible to predict the sales of a product type eight weeks in advance, however, this will need to be confirmed using other product types.

The second approach to extract value from Twitter is to see what a continuous feed of messages contains. As a first step, we visualize the obtained tweets per product in a top-10 application. The main page of this application shows the top 10 most discussed products on Twitter in the last day for both fruits/vegetables and plants/flowers. By clicking on one of the products in these top lists, we are redirected to a page, shown in Figure 1 for bananas, which shows us both the current messages mentioning the product and a detailed analysis of these messages, e.g., in terms of the most occurring terms and a time series in which the terms are mentioned. Besides knowing what products are mentioned frequently, we also use the real-time data for the detection of stories and discussions that suddenly pop-up. We do this by clustering incoming tweets by their tokens, using the Jaccard index [Jaccard, 1901]. If the tokens of two tweets are more similar than a predefined threshold, which we set at 0.6, then these two tweets will be represented by the same cluster. Therefore, if a topic is actively discussed on Twitter, it will be represented as a cluster in our story detection. Since these clusters are renewed every hour, we add the notion of stories, which clusters the clusters over time. By doing this, we can also track which clusters are prevalent for a longer period of time and therefore will be very likely to be of value for the industry.

In this paper, we described our first steps towards empowering the horticulture industry by analyzing topic-relevant tweets in Twitter. During our first exploration of the Twitter data, we found that there could be predictive power in the number of times a specific product is mentioned on Twitter for the future sales numbers of that particular product. Furthermore, we developed methods to visualize the current industry-specific content in real-time and filter out interesting information in the process. These ideas can be fruitfully adopted in marketing analytics to directly measure the impact of marketing activities. These first results provide a good basis for further study.

[2] `https://dev.twitter.com/streaming/reference/post/statuses/filter`

[3] `https://dev.twitter.com/rest/public/rate-limits`

References

Bhattacharya, D., & Ram, S. (2012). Sharing news articles using 140 characters: A diffusion analysis on Twitter. *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on* (pp. 966–971).

Castillo, C., El-Haddad, M., Pfeffer, J., & Stempeck, M. (2014). Characterizing the life cycle of online news stories using social media reactions. *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (pp. 211–223). New York, NY, USA: ACM.

Fan, W., & Gordon, M. D. (2014). The power of social media analytics. *Commun. ACM*, *57*, 74–81.

Hays, S., Page, S. J., & Buhalis, D. (2013). Social media as a destination marketing tool: its use by national tourism organisations. *Current Issues in Tourism*, *16*, 211–239.

He, W., Wu, H., Yan, G., Akula, V., & Shen, J. (2015). A novel social media competitive analytics framework with sentiment benchmarks. *Information & Management*, *52*, 801–812. Novel applications of social media analytics.

Jaccard, P. (1901). Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bull. Soc. Vaud. Sci. Nat.*, *37*, 241–272.

Järvinen, J., Töllmen, A., & Karjaluoto, H. (2015). *"marketing dynamism & sustainability: Things change, things stay the same. . . "*, chapter "Web Analytics and Social Media Monitoring in Industrial Marketing: Tools for Improving Marketing Communication Measurement", 477–486. Springer International Publishing.

Lanagan, J., & Smeaton, A. F. (2011). Using Twitter to Detect and Tag Important Events in Live Sports. *Artificial Intelligence*, *29*, 542–545.

Lehmann, J., Castillo, C., Lalmas, M., & Zuckerman, E. (2013). Transient news crowds in social media. *Proceedings of the Conference on Weblogs and Social Media* (pp. 351–360).

Nichols, J., Mahmud, J., & Drews, C. (2012). Summarizing sporting events using twitter. *IUI '12: Proceedings of the 2012 ACM international conference on Intelligent User Interfaces* (pp. 189–198).

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, *58*, 240–242.

ten Thij, M., Bhulai, S., van den Berg, W., & Zwinkels, H. (2016). Twitter Analytics for the Horticulture Industry. *International Conference on DATA ANALYTICS 2016* (pp. 75–79). Venice, Italy: IARIA.

Van Oorschot, G., Van Erp, M., & Dijkshoorn, C. (2012). Automatic extraction of soccer game events from Twitter. *CEUR Workshop Proceedings* (pp. 21–30).

Xu, C., Yu, Y., & Hoi, C.-K. (2015). Hidden in-game intelligence in NBA players' tweets. *Commun. ACM*, *58*, 80–89.

Graph-based semi-supervised learning for complex networks

Leto Peel LETO.PEEL@UCLOUVAIN.BE
ICTEAM, Université Catholique de Louvain, Louvain-la-Neuve, Belgium
naXys, Université de Namur, Namur, Belgium

Keywords: semi-supervised learning, complex networks, classification

Abstract

We address the problem of semi-supervised learning in relational networks, networks in which nodes are entities and links are the relationships or interactions between them. Typically this problem is confounded with the problem of graph-based semi-supervised learning (GSSL), because both problems represent the data as a graph and predict the missing class labels of nodes. However, not all graphs are created equally. In GSSL a graph is constructed, often from independent data, based on similarity. As such, edges tend to connect instances with the same class label. Relational networks, however, can be more heterogeneous and edges do not always indicate similarity. In this work (Peel, 2017) we present two scalable approaches for graph-based semi-supervised learning for the more general case of relational networks. We demonstrate these approaches on synthetic and real-world networks that display different link patterns within and between classes. Compared to state-of-the-art baseline approaches, ours give better classification performance and do so without prior knowledge of how classes interact.

In most complex networks, nodes have attributes, or metadata, that describe a particular property of the node. In some cases these attributes are only partially observed for a variety of reasons e.g. the data is expensive, time-consuming or difficult to accurately collect. In machine learning, classification algorithms are used to predict discrete node attributes (which we refer to as class labels) by learning from a training set of labelled data, i.e. data for which the target attribute values are known. Semi-supervised learning is a classification problem that aims to make use of both the unlabelled data and the labelled data typically used to train supervised models. A common approach is graph-based semi-supervised learning (GSSL) (Belkin & Niyogi, 2004), (Joachims, 2003), (Talukdar & Crammer, 2009), (Zhou et al., 2003), (Zhu et al., 2003), in which (often independent) data are represented as a *similarity graph*, such that a vertex is a data instance and an edge indicates similarity between two instances. By utilising the graph structure, of labelled and unlabelled data, it is possible to accurately classify the unlabelled vertices using a relatively small set of labelled instances.

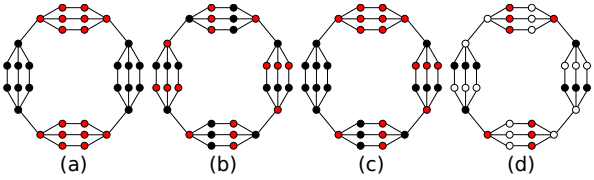


Figure 1. Different patterns of links between class labels {red, black}: (a) nodes with the same label tend to be linked (*assortative*), (b) links connect nodes with different labels (*link-heterogeneity*), (c) some nodes are assortative and some are not (*class-heterogeneity*), (d) missing labels (white) obscures the pattern of links.

Here we consider the semi-supervised learning problem in the context of *complex networks*. These networks consist of nodes representing entities (e.g. people, user accounts, documents) and links representing pairwise dependencies or relationships (e.g. friendships, contacts, references). Here class labels are discrete-valued attributes (e.g. gender, location, topic) that describe the nodes and our task is to predict these labels based only on the network structure and a small subset of nodes already labelled. This problem of classifying nodes in networks is often treated as a GSSL prob-

Preliminary work. Under review for Benelearn 2017. Do not distribute.

lem because the objective, to predict missing node labels, and the input, a graph, are the same. Sometimes this approach works well due to assortative mixing, or homophily, a feature frequently observed in networks, particularly in social networks. Homophily is the effect that linked nodes share similar properties or attributes and occurs either through a process of selection or influence. However, not all node attributes in complex networks are assortative. For example, in a network of sexual interactions between people it is likely that some attributes will be common across links, e.g. similar demographic information or shared interests, but other attributes will be different, e.g. links between people of different genders. Furthermore, the pattern of similarity or dissimilarity of attributes across links may not be consistent across the whole network, e.g. in some parts of the network links will occur between people of the same gender.

In situations where we have a sparsely labelled network and do not know the pattern of interaction between nodes of different classes, the problem of predicting the class labels of the remaining nodes is hard. Figure 1 shows a toy example in which nodes are assigned red or black labels and Fig. 1(a)–(c) show possible arrangements of labels that become indistinguishable if certain labels are missing (Fig. 1(d)). Tasks such as fraud detection face this type of problem, where certain patterns of interaction are indicative of nefarious behaviour (e.g. in communication (Cortes et al., 2002) or online auction (Chau et al., 2006) networks) but only a sparse set of confirmed fraudulent or legitimate users are available and no knowledge of how fraudsters operate or if there are different types of fraudulent behaviour.

In this work (Peel, 2017), we present two novel methods to deal with the problem of semi-supervised learning in complex networks. Both methods approximate equivalence relations from social network theory to define a notion of similarity that is robust to different patterns of interaction. We use these measures of similarity to implicitly construct similarity graphs from complex networks upon which we can propagate class label information. We demonstrate on synthetic networks that our methods are capable of classifying nodes under a range of different interaction patterns in which standard GSSL methods fail. Finally, we demonstrate on real data that our *two-step label propagation* approach performs consistently well against baseline approaches and easily scales to large networks with $O(10^6)$ nodes and $O(10^7)$ edges.

Acknowledgments

The author was supported by IAP DYSCO of the Belgian Scientific Policy Office, and ARC Mining and Optimization of Big Data Models of the Federation Wallonia-Brussels.

References

Belkin, M., & Niyogi, P. (2004). Semi-supervised learning on riemannian manifolds. *Machine learning*, *56*, 209–239.

Chau, D. H., Pandit, S., & Faloutsos, C. (2006). Detecting fraudulent personalities in networks of online auctioneers. *Lect. Notes Comput. Sc.*, *4213*, 103–114.

Cortes, C., Pregibon, D., & Volinsky, C. (2002). Communities of interest. *Intell. Data Anal.*, *6*, 211–219.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. *Int. Conf. Machine Learning* (pp. 290–297).

Peel, L. (2017). Graph-based semi-supervised learning for relational networks. *SIAM International Conference on Data Mining (SDM).* arXiv preprint arXiv:1612.05001.

Talukdar, P. P., & Crammer, K. (2009). New regularized algorithms for transductive learning. *Lect. Notes Comput. Sc.*, *5782*, 442–457.

Zhou, D., et al. (2003). Learning with local and global consistency. 321–328.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. *Int. Conf. on Machine Learning* (pp. 912–919).

Contact Patterns, Group Interaction and Dynamics on Socio-Behavioral Multiplex Networks

Martin Atzmueller M.ATZMULLER@UVT.NL
Tilburg University (TiCC), Warandelaan 2, 5037 AB Tilburg, The Netherlands

Lisa Thiele LISA.THIELE@TU-BRAUNSCHWEIG.DE
TU Braunschweig, Institute of Psychology Braunschweig, Germany

Gerd Stumme STUMME@CS.UNI-KASSEL.DE
University of Kassel (ITeG), Wilhelmshöher Allee 73, 34121 Kassel, Germany

Simone Kauffeld S.KAUFFELD@TU-BRAUNSCHWEIG.DE
TU Braunschweig, Institute of Psychology Braunschweig, Germany

Keywords: social network analysis, temporal dynamics, offline social networks, behavioral networks

Abstract

The analysis of social interaction networks is essential for understanding and modeling network structures as well as the behavior of the involved actors. This paper summarizes an analysis at large scale using (sensor) data collected by RFID tags complemented by self-report data obtained using surveys. We focus on the social network of a students' freshman week, and investigate research questions concerning group behavior and structure, gender homophily, and interrelations of sensor-based (RFID) and self-report social networks. Such analyses are a first step for enhancing interactions and enabling proactive guidance.

1. Introduction

The analysis of group interaction and dynamics is an important task for providing insights into human behavior. Based on the social distributional hypothesis (Mitzlaff et al., 2014) stating that users with similar interaction characteristics tend to be semantically related, we investigate such interaction networks, and analyze the respective relations. Social media and mobile devices allow the collection of interaction data at large scale, e. g., Bluetooth-enabled mobile phone data (Atzmueller & Hilgenberg, 2013), or Radio Frequency Identification (RFID) devices (Barrat et al., 2008). However, the combination of both sources is used rather seldomly so far.

This paper summarizes an analysis of social interactions on networks of face-to-face proximity complemented by self-report data in the context of that students' freshman week presented in (Atzmueller et al., 2016b). This freshman week, the first week of freshman students at a psychology degree program, is organized as a special course (five days) before the regular courses start. We collected two types of network data: Person-to-person interaction using self-report questionnaires and active RFID (radio frequency identification) tags with proximity sensing, cf. (Barrat et al., 2008). We focus on structural and dynamic behavioral aspects as well as on properties of the participants, i. e., gender homophily. Furthermore, we investigate the relation of social interaction networks of face-to-face proximity and networks based on self-reports, extending the analysis in (Thiele et al., 2014).

Summarizing our results, we show that there are distinctive structural and behavioral patterns in the face-to-face proximity network corresponding to the activities of the freshman week. Specifically, we analyze the evolution of contacts, as well as the individual connectivity according to the phases of the event. Furthermore, we show the influence of gender homophily on the face-to-face proximity activity.

Appearing in *Proceedings of Benelearn 2017*.

2. Related Work

The SocioPatterns collaboration developed an infrastructure that detects close-range and face-to-face proximity (1-1.5 meters) of individuals wearing proximity tags with a temporal resolution of 20 seconds (Cattuto et al., 2010). In contrast to, e. g., bluetooth-based methods that allow the analysis based on co-location data (Atzmueller & Hilgenberg, 2013), here face-to-face proximity can be observed with a probability of over 99% using the interval of 20 seconds for a minimal contact duration. This infrastructure has been deployed in various environments for studying the dynamics of human contacts, e. g., conferences (Cattuto et al., 2010; Atzmueller et al., 2012; Macek et al., 2012), workplaces (Atzmueller et al., 2014a), or schools (Mastrandrea et al., 2015).

The analysis of interaction and groups, and their evolution, respectively, are prominent topics in social sciences, e. g., (Turner, 1981; Atzmueller et al., 2014b). The temporal evolution of contact networks and induced communities is analyzed, for example, in (Barrat & Cattuto, 2013; Kibanov et al., 2014). Also, the evolution of social groups has been investigated in a community-based analysis (Palla et al., 2007) using bibliographic and call-detail records. Furthermore, the analysis of link relations and their prediction is investigated in, e. g., (Liben-Nowell & Kleinberg, 2003; Christoph Scholz and Martin Atzmueller and Alain Barrat and Ciro Cattuto and Gerd Stumme, 2013). Overall, social interaction networks in online and offline contexts, important features, as well as methods for analysis are summarized in (Atzmueller, 2014).

In contrast to the approaches above, this paper focuses on networks of face-to-face proximity (F2F) at a students' freshman week, combining RFID-based networks of a newly composed group with networks obtained by self-reports (SRN). To the best of the authors' knowledge, this is the first time that such an analysis has been performed using real-world networks of face-to-face proximity of a newly composed group together with the corresponding questionnaire data.

3. Dataset

The dataset contains data from 77 students (60 females and 17 males) attending the freshman week. We asked each student to wear an active RFID tag while they were staying at the facility. The RFID deployment at the freshman week utilized a variant of the MyGroup (Atzmueller et al., 2014a) system for data collection. Participants volunteered to wear active RFID proximity tags, which can sense and log the close-range face-to-face proximity of individuals wearing them.

4. Results and Future Work

We analyze data of a students' freshman week and show that there are distinctive structural patterns in the F2F data corresponding to the activities of the freshman week. This concerns both the static structure as well as its dynamic evolution of contacts and the individual connectivity in the network according to the individual phases of the event. Furthermore, we show the effects of gender homophily on the contact activity. Finally, our results also indicate existing structural associations between the face-to-face proximity network and various self-report networks. In the context of introductory courses, this points out the importance of stronger ties (long conversations) between the students at the very beginning of their studies for fostering an easier start, better cooperativeness and support between the students. Our results especially show the positive effect of the freshman week for supporting the connectivity between students; the analysis also indicates the benefit of such a course of five days with respect to the interaction and contact patterns in contrast to shorter introductory courses. Such insights into contact patterns and their dynamics enable design and modeling decision support for organizing such events and for enhancing interaction of its participants, e. g., considering group organization, recommendations, notifications, and proactive guidance.

For future work, we aim to analyze structure and semantics (Mitzlaff et al., 2011; Mitzlaff et al., 2014) further, e. g., in order to investigate, if different network data can be predicted, e. g., (Scholz et al., 2012; Christoph Scholz and Martin Atzmueller and Alain Barrat and Ciro Cattuto and Gerd Stumme, 2013). For that, also multiplex networks, e. g., based on co-location proximity information (Scholz et al., 2011) can be applied. Here, subgroup discovery and exceptional model mining, e. g., (Leman et al., 2008; Atzmueller, 2015) provide interesting approaches, especially when combining compositional and structural analysis, i. e., on attributed graphs (Atzmueller et al., 2016a; Atzmueller, 2016). Furthermore, we aim to integrate our results into smart approaches, e. g., as enabled by augmenting the Ubicon platform (Atzmueller et al., 2014a) also including explanation-aware methods (Atzmueller & Roth-Berghofer, 2010). Potential goals include enhancing interactions at such events, as well as to support the organization of such events concerning group composition, and the setup of activities both at the micro- and macro-level. Developing suitable recommendation, notification, and proactive guidance systems that are triggered according to the events structure and dynamics are further directions for future work.

References

Atzmueller, M. (2014). Data Mining on Social Interaction Networks. *JDMDH*, *1*.

Atzmueller, M. (2015). Subgroup Discovery – Advanced Review. *WIREs DMKD*, *5*, 35–49.

Atzmueller, M. (2016). Detecting Community Patterns Capturing Exceptional Link Trails. *Proc. IEEE/ACM ASONAM*. IEEE Press.

Atzmueller, M., Becker, M., Kibanov, M., Scholz, C., Doerfel, S., Hotho, A., Macek, B.-E., Mitzlaff, F., Mueller, J., & Stumme, G. (2014a). Ubicon and its Applications for Ubiquitous Social Computing. *New Review of Hypermedia and Multimedia*, *20*, 53–77.

Atzmueller, M., Doerfel, S., Hotho, A., Mitzlaff, F., & Stumme, G. (2012). *Face-to-Face Contacts at a Conference: Dynamics of Communities and Roles*, vol. 7472 of *LNAI*. Springer.

Atzmueller, M., Doerfel, S., & Mitzlaff, F. (2016a). Description-Oriented Community Detection using Exhaustive Subgroup Discovery. *Information Sciences*, *329*, 965–984.

Atzmueller, M., Ernst, A., Krebs, F., Scholz, C., & Stumme, G. (2014b). On the Evolution of Social Groups During Coffee Breaks. *Proc. WWW 2014 (Companion)* (pp. 631–636). ACM.

Atzmueller, M., & Hilgenberg, K. (2013). Towards Capturing Social Interactions with SDCF: An Extensible Framework for Mobile Sensing and Ubiquitous Data Collection. *Proc. MSM 2013*. ACM Press.

Atzmueller, M., & Roth-Berghofer, T. (2010). The Mining and Analysis Continuum of Explaining Uncovered. *Proc. AI-2010*. London, UK: SGAI.

Atzmueller, M., Thiele, L., Stumme, G., & Kauffeld, S. (2016b). Analyzing Group Interaction and Dynamics on Socio-Behavioral Networks of Face-to-Face Proximity. *Proc. ACM Ubicomp Adjunct*. ACM.

Barrat, A., & Cattuto, C. (2013). *Temporal Networks*, chapter Temporal Networks of Face-to-Face Human Interactions. Understanding Complex Systems. Springer.

Barrat, A., Cattuto, C., Colizza, V., Pinton, J.-F., den Broeck, W. V., & Vespignani, A. (2008). High Resolution Dynamical Mapping of Social Interactions with Active RFID. *PLoS ONE*, *5*.

Cattuto, C., Van den Broeck, W., Barrat, A., Colizza, V., Pinton, J.-F., & Vespignani, A. (2010). Dynamics of Person-to-Person Interactions from Distributed RFID Sensor Networks. *PLoS ONE*, *5*.

Christoph Scholz and Martin Atzmueller and Alain Barrat and Ciro Cattuto and Gerd Stumme (2013). New Insights and Methods For Predicting Face-To-Face Contacts. *Proc. ICWSM*. AAAI Press.

Kibanov, M., Atzmueller, M., Scholz, C., & Stumme, G. (2014). Temporal Evolution of Contacts and Communities in Networks of Face-to-Face Human Interactions. *Sci Chi Information Sciences*, *57*.

Leman, D., Feelders, A., & Knobbe, A. (2008). Exceptional Model Mining. *Proc. ECML-PKDD* (pp. 1–16). Berlin: Springer.

Liben-Nowell, D., & Kleinberg, J. M. (2003). The Link Prediction Problem for Social Networks. *Proc. CIKM* (pp. 556–559). ACM.

Macek, B.-E., Scholz, C., Atzmueller, M., & Stumme, G. (2012). Anatomy of a Conference. *Proc. ACM Hypertext* (pp. 245–254). ACM.

Mastrandrea, R., Fournet, J., & Barrat, A. (2015). Contact Patterns in a High School: A Comparison between Data Collected Using Wearable Sensors, Contact Diaries and Friendship Surveys. *PLoS ONE*, *10*.

Mitzlaff, F., Atzmueller, M., Benz, D., Hotho, A., & Stumme, G. (2011). Community Assessment using Evidence Networks. In *Analysis of Social Media and Ubiquitous Data*, vol. 6904 of *LNAI*, 79–98. Springer.

Mitzlaff, F., Atzmueller, M., Hotho, A., & Stumme, G. (2014). The Social Distributional Hypothesis. *Journal of Social Network Analysis and Mining*, *4*.

Palla, G., Barabasi, A.-L., & Vicsek, T. (2007). Quantifying Social Group Evolution. *Nature*, *446*, 664–667.

Scholz, C., Atzmueller, M., & Stumme, G. (2012). On the Predictability of Human Contacts: Influence Factors and the Strength of Stronger Ties. *Proc. SocialCom* (pp. 312–321). IEEE.

Scholz, C., Doerfel, S., Atzmueller, M., Hotho, A., & Stumme, G. (2011). Resource-Aware On-Line RFID Localization Using Proximity Data. *Proc. ECML-PKDD* (pp. 129–144). Springer.

Thiele, L., Atzmueller, M., Kauffeld, S., & Stumme, G. (2014). Subjective versus Objective Captured Social Networks: Comparing Standard Self-Report Questionnaire Data with Observational RFID Technology Data. *Proc. Measuring Behavior*. Wageningen, The Netherlands.

Turner, J. C. (1981). Towards a Cognitive Redefinition of the Social Group. *Cah Psychol Cogn*, *1*.

Deep Learning Track

Research Papers

Modeling brain responses to perceived speech with LSTM networks

Julia Berezutskaya JU.BEREZUTSKAYA@GMAIL.COM
Zachary V. Freudenburg Z.V.FREUDENBURG@UMCUTRECHT.NL
Nick F. Ramsey N.F.RAMSEY@UMCUTRECHT.NL
Brain Center Rudolf Magnus, Department of Neurology and Neurosurgery, University Medical Center Utrecht, Heidelberglaan 100, 3584 CX, Utrecht, The Netherlands

Umut Güçlü U.GUCLU@DONDERS.RU.NL
Marcel A.J. van Gerven M.VANGERVEN@DONDERS.RU.NL
Radboud University, Donders Institute for Brain, Cognition and Behaviour, Montessorilaan 3, 6525 HR, Nijmegen, The Netherlands

Keywords: LSTM, RNN, brain responses, speech

Abstract

We used recurrent neural networks with long-short term memory units (LSTM) to model the brain responses to speech based on the speech audio features. We compared the performance of the LSTM models to the performance of the linear ridge regression model and found the LSTM models to be more robust for predicting brain responses across different feature sets.

1. Introduction

One of the approaches to understanding how the human brain processes information is through modeling the observed neural activity evoked during an experimental task. Typically, the neural activation data are collected as a response to a set of stimuli, for example pictures, audio or video clips. Then, salient features are extracted from the stimulus set and used to model the neural responses. The learned mapping is called a neural encoding model (Kay et al., 2008; Naselaris et al., 2012).

A common approach is to use hand-engineered features, which can be complex transformations of the stimulus set and learn a linear mapping between the stimulus features and the neural responses. In case of speech, the spectrogram and non-linear spectrotemporal modulation features have been used in linear encoding models (Santoro et al., 2014).

Non-linear models of neural encoding have recently started to gain popularity in the neuroscience community, since they allow learning a more complex mapping between the stimulus features and the neural responses. In a recent study, various models from the recurrent neural network family were trained to predict the neural responses to video clips (Güçlü & van Gerven, 2017).

In the present study, we apply LSTM models to predict the neural responses to continuous speech. We use various sets of stimulus features for model training and compare the performance of the LSTM models with the performance of a linear encoding model.

2. Methods

Brain data collection and preprocessing

Fifteen patients with medication-resistant epilepsy underwent implantation of subdural electrodes (electrocorticography, ECoG). All patients gave written consent to participate in research tasks alongside the clinical procedures to determine the source of the epileptic activity. During the research task, the patients watched a 6.5 min short movie with a coherent plot (fragments of Pippi Longstocking, 1969) while their neural activity was recorded through the ECoG electrodes. The ECoG recordings were acquired with a 128 channel recording system (Micromed, Treviso, Italy) at a sampling rate (SR) of 512 Hz filtered at 0.15-134.4 Hz. All patients had electrodes in temporal and frontal cortices, implicated in auditory and language

Preliminary work. Under review for Benelearn 2017. Do not distribute.

processing (Howard et al., 2000; Hickok & Poeppel, 2007; Friederici, 2012; Kubanek et al., 2013).

The collected ECoG data were preprocessed prior to model fitting. Per patient, based on the visual inspection, electrodes with noisy or flat signal were excluded from the dataset. Notch filter at 50 and 100 Hz was used to remove line noise and common average re-referencing was applied. The Gabor wavelet decomposition was used to extract neural responses in the high frequency band (HFB, 60-120 Hz) from the time domain signal. The Wavelet decomposition was applied in the HFB range in 1 Hz bins with decreasing window length (4 wavelength full-width at half max). The resulting signal was averaged over the whole range to produce a single HFB neural response per electrode. The resulting neural responses were downsampled to 125 Hz. The preprocessed data were concatenated across patients over the electrode dimension (total number of electrodes = 1283).

Audio features

The soundtrack of the movie contained speech and music fragments. From the soundtrack, we constructed three input feature sets for training the models. First, we extracted the waveform of the movie soundtrack and downsampled it to 16000 Hz. To create the first, time-domain, feature set (*time*), we reshaped the waveform to the matrix of size $N \times F_1$, where N is the number of time points at the SR of the neural responses (125 Hz), and F_1 is 128 time features (16000/125) To make the second feature set, we extracted a sound spectrogram at 128 logarithmically spaced bins in range 180-7000 Hz. This resulted in a $N \times F_2$ matrix with $F_2 = 128$ features (*freq*). Finally, the spectrogram was filtered with a bank of 2D Gabor filters to extract spectrotemporal modulation energy features (Chi et al., 2005). The filtering was done at 16 logarithmically spaced bins in range 0.25-40 Hz along the temporal dimension, and 8 logarithmically spaced bins in range 0.25-4 cyc/oct along the frequency dimension. The third feature matrix $N \times F_3$ was built by concatenating all spectrotemporal modulation features: 16×8, $F_3 = 128$ features (*smtm*). The spectrogram and the spectrotemporal modulation energy features were obtained using the NSL toolbox (Chi et al., 2005).

Linear encoding model

For each input feature set, a separate kernel linear ridge regression (Murphy, 2012) was trained to predict the neural responses to speech fragments. The HFB neural response of each electrode y_e at time point t was modeled as a linear combination of the input audio features at this time point:

$$y_e(t) = \boldsymbol{\beta}_e{}^\top \mathbf{x}(t) + \epsilon_e$$

where $\epsilon_e \sim \mathcal{N}(0, \sigma^2)$.

L^2 penalized least squares loss function was analytically minimized to estimate the regression coefficients $\boldsymbol{\beta}_e$. The kernel trick was used to avoid large matrix inversions in the input feature space:

$$\boldsymbol{\beta}_e = \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top + \lambda_e \mathbf{I}_n)^{-1} \mathbf{y}_e$$

where n is the number of training time points.

A nested cross-validation was used to estimate the amount of regularization λ_e (Güçlü & van Gerven, 2014). First, a grid of the effective degrees of freedom of the model fit was specified. Then, Newton's method was used to solve the effective degrees of freedom for λ_e. Finally, λ_e that resulted in the lowest nested cross-validation error was taken as the final estimate.

The model was tested on 5% of all data. A five-fold cross-validation was used to validate the model performance. In each cross-validation fold different speech fragments were selected for testing, so that no data points were shared in test sets across five folds.

Model performance was measured as the Spearman correlation between predicted and observed neural responses in the test set. The correlation values were averaged across five cross-validation folds and were transformed to t-values for determining significance (Kendall & Stuart, 1961).

LSTM encoding models

For each input feature set, six LSTM models (Hochreiter & Schmidhuber, 1997) with varying architectures were trained to predict the neural responses to speech fragments. The six LSTM models were specified using a varying number of hidden layers (one or two) and a varying number of units per hidden layer (20, 50 or 100). A fully-connected linear layer was specified as the output layer. The neural response of each electrode y_e at time point t was modeled as a linear combination of the hidden states $\mathbf{h}(t)$. For models with one hidden LSTM layer (*1-lstm20, 1-lstm50, 1-lstm100*):

$$y_e(t) = \boldsymbol{\beta}_e^\top \mathbf{h_1}(t) + b_e + \epsilon_e$$

where b_e is a bias and $\epsilon_e \sim \mathcal{N}(0, \sigma^2)$.

For models with two hidden LSTM layers (*2-lstm20, 2-lstm50, 2-lstm100*):

$$y_e(t) = \boldsymbol{\beta}_e^\top \mathbf{h}_2(t) + b_e + \epsilon_e$$

The hidden states $\mathbf{h}_1(t)$ were computed in the following way:

$$\begin{aligned}
\mathbf{f}(t) &= \sigma\left(\mathbf{U}_f\mathbf{h}_1(t-1) + \mathbf{W}_f\mathbf{x}(t) + \mathbf{b}_f\right) \\
\mathbf{i}(t) &= \sigma\left(\mathbf{U}_i\mathbf{h}_1(t-1) + \mathbf{W}_i\mathbf{x}(t) + \mathbf{b}_i\right) \\
\mathbf{o}(t) &= \sigma\left(\mathbf{U}_o\mathbf{h}_1(t-1) + \mathbf{W}_o\mathbf{x}(t) + \mathbf{b}_o\right) \\
\mathbf{c}(t) &= \mathbf{i}(t)\tanh\left(\mathbf{U}_c\mathbf{h}_1(t-1) + \mathbf{W}_c\mathbf{x}(t) + \mathbf{b}_c\right) \\
&\quad + \mathbf{f}(t)\mathbf{c}(t-1) \\
\mathbf{h}_1(t) &= \mathbf{o}(t)\tanh\left(\mathbf{c}(t)\right)
\end{aligned}$$

where σ is the logistic sigmoid function. Vectors $\mathbf{f}(t)$, $\mathbf{i}(t)$, $\mathbf{o}(t)$ and $\mathbf{c}(t)$ correspond to four LSTM gates: *forget gate*, *input gate*, *output gate* and *cell state*, respectively. Matrices $\mathbf{U}$ and $\mathbf{W}$ contain the gate-specific weights and vectors $\mathbf{b}$ are the gate-specific bias vectors.

For models with two hidden LSTM layers (*2-lstm20, 2-lstm50, 2-lstm100*), the hidden states $\mathbf{h}_2(t)$ were computed in the similar way, except that the input to the cells at the second layer was $\mathbf{h}_1(t)$.

Mean squared error function was minimized during the training using Adam optimizer (Kingma & Ba, 2014). The models were trained using backpropagation through time, with truncation after the i-th time point, corresponding to the 500 ms lag. Each model was optimized using a validation set (5% of all data) and early stopping: training was stopped if the loss on the validation set did not decrease for 30 epochs. The model with least loss on validation set was used as the final model. Each model was tested on 5% of all data. Chainer package (Tokui et al., 2015) was used for implementing the LSTM models.

The model performance was computed in the same way as for the linear model. Similarly, a five-fold cross-validation was used to validate the model performance. The correlations were transformed to t-values for determining significance.

Model performance comparison

For each model, the model performance scores (Spearman correlations) were thresholded at $p < .001$, Bonferroni corrected for the number of electrodes. Per each feature set, we selected the electrodes with significant performance across all the models: 53 electrodes for *freq*, 125 electrodes for *smtm* and 0 for *time*. The performance across the models was compared using one-way ANOVA test. Tuckey's honest significant difference (HSD) test was used post-hoc to determine pairs of models with significantly different mean performance values. Separately, per each model, we calculated the number of electrodes for which the model achieved significant performance.

3. Results

When trained on *time*, the performance of the linear ridge regression model was not significant $p < .001$, Bonferroni corrected. All the LSTM models performed significantly above chance. When trained on *freq*, there was significant difference in performance between the linear ridge regression model and the LSTM models ($F(1119) = 12.65, p = 5.69 \times 10^{-13}$, Tuckey's HSD test: each pair of ridge−LSTM means were significantly different at $p = 0.001$). When trained on *smtm*, there was no significant difference between the linear ridge regression model and the LSTM models ($F(874) = 1.7, p = .12$). Overall, the LSTM models showed good performance with all three feature sets (Fig. 1). The performance of the linear ridge regression model depended strongly on the input feature set and improved as the input features became more complex. Despite varying the parameters of the LSTM architecture, there was almost no difference in performance among the six LSTM models. We observed a significant difference in LSTM model performance for the *time* feature set: $F(275) = 9.37, p = 2.99 \times 10^{-8}$. For both one- and two-layer LSTMs, the models with 20 hidden units performed worse compared to the models with a larger number of hidden units (based on the HSD test).

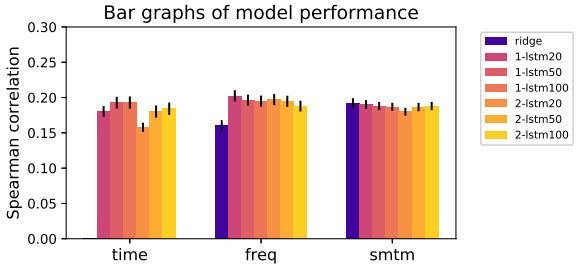


Figure 1. Model performance comparison between the linear ridge regression model and the six LSTM models, trained on separate feature sets: *time*, *freq* and *smtm*. The bars show mean model performance scores over the electrodes (Spearman correlations). The scores were significant at $p < .001$, Bonferroni corrected for the number of electrodes. Error bars indicate standard error of the mean.

All models trained on all feature sets performed significantly above chance only in a subset of all electrodes. For all feature sets, the LSTM models achieved significant performance in a larger amount of electrodes, compared to the linear ridge regression model (Table 1).

All models but the linear ridge regression model

Table 1. Percentage of electrodes the models performed well for when trained on each feature set. Total number of electrodes (100%) is 1283. Highest values are in bold.

MODEL	TIME	FREQ	SMTM
RIDGE	0%	6%	16%
1-LSTM20	**8**%	10%	19%
1-LSTM50	**9**%	**12**%	**25**%
1-LSTM100	7%	**12**%	**28**%
2-LSTM20	7%	11%	17%
2-LSTM50	5%	**12**%	23%
2-LSTM100	**8**%	**12**%	**26**%

trained on *time*, showed significant performance in electrodes located in the temporal cortex (superior temporal gyrus), implicated in auditory processing (Howard et al., 2000; Norman-Haignere et al., 2015). The LSTM models trained on *time* performed significantly well for the electrodes in the superior temporal gyrus. The LSTM models trained on *freq* and *smtm* showed involvement of the electrodes located in the frontal cortex, as well as the posterior middle temporal and parietal cortices (Fig. 2). These cortical regions are implicated in language and other higher-level cognitive processing (Hagoort, 2013; Friederici, 2012).

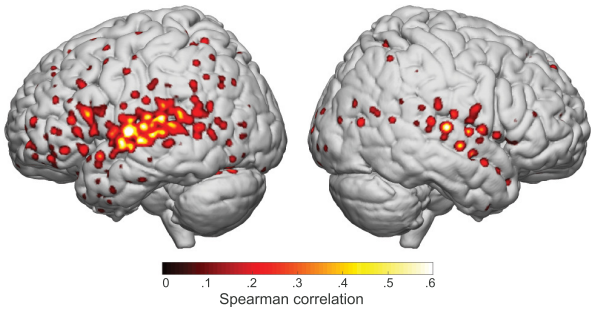


Figure 2. Cortical locations of the electrodes whose responses were modeled significantly above chance (at $p < .001$, Bonferroni corrected for the number of electrodes) by 1-LSTM50 trained on *smtm* feature set.

4. Discussion

In the present study we trained several models to predict the neural responses to perceived speech. The neural responses were obtained using ECoG. We considered a linear ridge regression model and recurrent neural network models with LSTM units varying in architecture. Each model was trained on three separate sets of audio features. We found that the performance of the linear ridge regression model depended strongly on the set of the input features. Notably, the linear ridge regression model did non achieve significant performance using the time domain features. In contrast, the LSTM models showed comparable performance across different feature sets. Using more complex audio features allowed the LSTM models to make accurate predictions for a larger set of ECoG electrodes.

There are multiple reasons why the linear ridge regression model and the LSTM models might have shown different performance when trained on *time* and *freq*. For example, the linear ridge regression model was regularized as opposed to the LSTM models presented here. Additionally, we retrained the LSTM models using a weight decay parameter for regularizing the network weights. The amount of the weight decay was cross-validated, but in multiple cases its optimal value turned out to be zero, and the overall performance of the LSTM models did not change considerably.

Other factors contributing to the superior performance of the LSTM models include presence of non-linear transformations within the LSTM cells (σ and tanh), as well as the *cell states* $\mathbf{c}$ which accumulate the information relevant for the predictions over time. Finally, the linear ridge regression model and the LSTM models differed considerably with respect to the number of the free parameters. We found it challenging to match the linear regression and neural network models with respect to all mentioned issues. Further work is necessary to determine which concrete properties of the LSTM models allowed it to outperform the linear ridge regression model when trained on *time* and *freq*.

The present work has a number of limitations. Because the placement of the ECoG grids varies across patients (depending on the tentative source of epilepsy), it is usually challenging to generalize the model performance to new patients' data. Here we used data from all patients to train the models. The model performance was then cross-validated using a five-fold cross-validation. Increasing the amount of patients could provide a larger overlap in the location of the electrodes. Then, a generalization to the data of unseen patients could be attempted.

5. Conclusions and future work

We trained several LSTM models to predict neural responses to speech based on the speech audio features and compared it to the performance of the linear ridge regression model. In general, the performance of the LSTM models was superior to the performance of the

linear ridge regression model in terms of the prediction accuracy and the amount of electrodes the models were successfully fit for. Further work is planned to investigate in detail what factors contribute to the superior performance of the LSTM models, compared to the linear ridge regression model. Some work on exploring the internal representations learned by the LSTM models (*cell states*) is also planned. Finally, we intend to compare the performance of RNNs with the performance of a convolutional neural network, trained on the wavelet-decomposed audio signal to predict the brain responses.

Acknowledgments

The work was supported by the NWO Gravitation grant 024.001.006.

References

Chi, T., Ru, P., & Shamma, S. A. (2005). Multiresolution spectrotemporal analysis of complex sounds. *The Journal of the Acoustical Society of America*, *118*, 887–906.

Friederici, A. D. (2012). The cortical language circuit: from auditory perception to sentence comprehension. *Trends in cognitive sciences*, *16*, 262–268.

Güçlü, U., & van Gerven, M. A. (2014). Unsupervised feature learning improves prediction of human brain activity in response to natural images. *PLoS Comput Biol*, *10*, e1003724.

Güçlü, U., & van Gerven, M. A. (2017). Modeling the dynamics of human brain activity with recurrent neural networks. *Frontiers in Computational Neuroscience*, *11*, 10–3389.

Hagoort, P. (2013). Muc (memory, unification, control) and beyond. *Frontiers in Psychology*, *4*, 416.

Hickok, G., & Poeppel, D. (2007). The cortical organization of speech processing. *Nature Reviews Neuroscience*, *8*, 393–402.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–1780.

Howard, M. A., Volkov, I., Mirsky, R., Garell, P., Noh, M., Granner, M., Damasio, H., Steinschneider, M., Reale, R., Hind, J., et al. (2000). Auditory cortex on the human posterior superior temporal gyrus. *Journal of Comparative Neurology*, *416*, 79–92.

Kay, K. N., Naselaris, T., Prenger, R. J., & Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature*, *452*, 352–355.

Kendall, M. G., & Stuart, A. (1961). The advanced theory of statistics (vol. 2), london: Charles w. *Griffin and Co., Ltd*, 1959–1963.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, *abs/1412.6980*.

Kubanek, J., Brunner, P., Gunduz, A., Poeppel, D., & Schalk, G. (2013). The tracking of speech envelope in the human cortex. *PloS one*, *8*, e53398.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Naselaris, T., Stansbury, D. E., & Gallant, J. L. (2012). Cortical representation of animate and inanimate objects in complex natural scenes. *Journal of Physiology-Paris*, *106*, 239–249.

Norman-Haignere, S., Kanwisher, N. G., & McDermott, J. H. (2015). Distinct cortical pathways for music and speech revealed by hypothesis-free voxel decomposition. *Neuron*, *88*, 1281–1296.

Santoro, R., Moerel, M., De Martino, F., Goebel, R., Ugurbil, K., Yacoub, E., & Formisano, E. (2014). Encoding of natural sounds at multiple spectral and temporal resolutions in the human auditory cortex. *PLoS Comput Biol*, *10*, e1003412.

Tokui, S., Oono, K., Hido, S., & Clayton, J. (2015). Chainer: a next-generation open source framework for deep learning. *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

Towards unsupervised signature extraction of forensic logs.
http://wwwis.win.tue.nl/~benelearn2017

Stefan Thaler S.M.THALER@TUE.NL
TU Eindhoven, Den Dolech 12, 5600 MB Eindhoven, Netherlands

Vlado Menkovski V.MENKOVSKI@TUE.NL
TU Eindhoven, Den Dolech 12, 5600 MB Eindhoven, Netherlands

Milan Petković MILAN.PETKOVIC@PHILIPS.COM
Philips Research, High Tech Campus 34, Eindhoven, Netherlands
TU Eindhoven, Den Dolech 12, 5600 MB Eindhoven, Netherlands

Keywords: RNN auto-encoder, log signature extraction, representation learning, clustering

Abstract

Log signature extraction is the process of finding a set of templates generated a set of log messages from the given log messages. This process is an important pre-processing step for log analysis in the context of information forensics because it enables the analysis of event sequences of the examined logs. In earlier work, we have shown that it is possible to extract signatures using recurrent neural networks (RNN) in a supervised manner (Thaler et al., 2017). Given enough labeled data, this supervised approach works well, but obtaining such labeled data is labor intensive.

In this paper, we present an approach to address the signature extraction problem in an unsupervised way. We use an RNN auto-encoder to create an embedding for the log lines and we apply clustering in the embedded space to obtain the signatures.

We experimentally demonstrate on a forensic log that we can assign log lines to their signature cluster with a V-Measure of 0.94 and a Silhouette score of 0.75.

1. Introduction

System- and application logs track activities of users and applications on computer systems. Log messages in such logs commonly consist of at least a time stamp and a free text message. The log message's time stamp indicates when an event has happened, and the text message describes what has happened. Log messages contain relevant information about the state of the software or actions that have been performed on a system, which makes them an invaluable source of information for a forensic investigator.

Ideally, forensic investigators should be able to extract information from such logs in an automated fashion. However, extracting information in an automated way is difficult for four reasons. First, the text contents of log messages are not uniformly structured. Second, there are different types of log messages in a system log. Thirdly, the text content may consist of variable and constant parts. The variable parts may have arbitrary values and length. Finally, the types of log messages change with updates of software and operating systems.

One way to enable automated information extraction is by manually creating a parser for these logs. However, writing a comprehensive parser is complex and labor intensive for the same reasons that it is difficult to analyze them automatically. A solution would be to use a learning approach to extract the log signatures automatically. A log signature is the "template" that has been used to create a log message, and extracting log signatures is the task of finding the log signatures given a set of log messages. An example of a log signa-

ture '*Initializing cgroup subsys* ***%s***', where '%s' acts as a placeholder for a mutable part. This signature can be used to create log lines such as '*Initializing cgroup subsys* ***pid***' or '*Initializing cgroup subsys* ***io***'.

Currently, log signatures are extracted in different ways. First, there are rule-based approaches. In rule-based approaches, signatures are manually defined, for example by using regular expressions. Rule-based approaches tend to work well when applied on logs with a limited amount of signatures. Second, there are algorithmic approaches, which use custom algorithms to extract signatures from logs. These algorithms are commonly tailored to specific types of logs. Finally, in previous work, we showed that supervised RNNs can also be used to derive log signatures from forensic logs (Thaler et al., 2017).

Our work is inspired by recent advances in modeling natural language using neural networks (Le & Mikolov, 2014; Cho et al., 2014; Johnson et al., 2016). Since log lines are partially natural language, we assume that neural language models will also capture the inherent structure of log lines well.

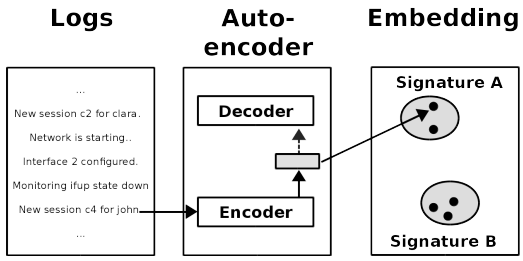


Figure 1. We first embed log lines using an RNN auto-encoder. We then cluster the embedded log lines to obtain the signatures.

Here, we propose an approach for addressing the signature extraction problem using attentive RNN auto-encoders. Figure 1 sketches our idea. The "encoder" transforms a log line into a dense vector representation, and the "decoder" attempts to reconstruct input log line in the reverse order. Log lines that belong to the same signature are embedded close to each other in the vector space. We then cluster the learned representations and use cluster centroids as signature descriptions.

The main contributions of this paper are:

- We present an approach for addressing the problem of extracting signatures from forensic logs. We are learning representations of log lines using an attentive recurrent auto-encoder. We detail this idea in Section 2.
- We provide first empirical evidence that this approach yields competitive results to state-of-the-art signature approaches. We detail our experiments in Section 3 and discuss the results in Section 4.

2. Signature extraction using attentive RNN auto-encoders

The main idea of our approach consists of two phases. In the first phase, we train an RNN auto-encoder to learn a representation of our log lines. To achieve this, we treat each log line as a sequence of words. This sequence serves as input to an RNN encoder, which encodes this sequence to a fixed, multi-dimensional vector. Based on this vector, the RNN decoder tries to reconstruct the reverse sequence of words. We detail this model in section 2. In the second phase, we cluster the encoded log lines based on their Euclidean distance to each other. We use the centroids of the clusters as signature description. We base this approach on the assumption that similar log lines are encoded close together in the embedding space. Intuitively, this assumption can be explained as follows. We let the model learn to reconstruct a log sequence from a fix-sized vector, which it previously encoded. Encoding a log line to a fixed-size vector is a sparsity constraint, which encourages the model to encode the log lines in distributed, dense manner. The more features such encoded log lines share, the closer to each other they will be in Euclidean space.

2.1. Model

Our model is based on the attentive RNN encoder-decoder architecture that was introduced by Bahdanau et al. (Dzmitry Bahdana et al., 2014) to address neural machine translation. We depict the schematic architecture in Figure 2. This model consists of three parts: an RNN encoder, an alignment model, and an RNN decoder.

We feed our model a sequence of n word ids $w_0 \dots w_n$. To retrieve the input word vectors for the RNN encoder we map each word to a unique vector x_i. This vector is represented by a row in a word embedding matrix $W^{v \times d}$ and the row is indexed by the position of the word in the vocabulary. v is the number of words in the vocabulary and d is a hyper parameter and represents the dimension of the embedding.

For a sequence of input word vectors $x_0 \dots x_i$ the RNN encoder outputs a sequence of output vectors $y_0 \dots y_n$

for input and a vector h that represents the encoded log line. h is the last hidden state of the RNN network.

The alignment model, also called attention mechanism, learns to weight the importance of the encoder's outputs for each decoding step. The output of the attention mechanism a context vector c_i that represents the weighted sum of the encoding outputs. This context vector is calculated for each decoding step. The alignment model increases the re-construction quality of the decoded sequences.

The decoders task is to predict the reversed input word sequence. It predicts the words for each time step, using the information of the encoded vector h and the context vector c_i.

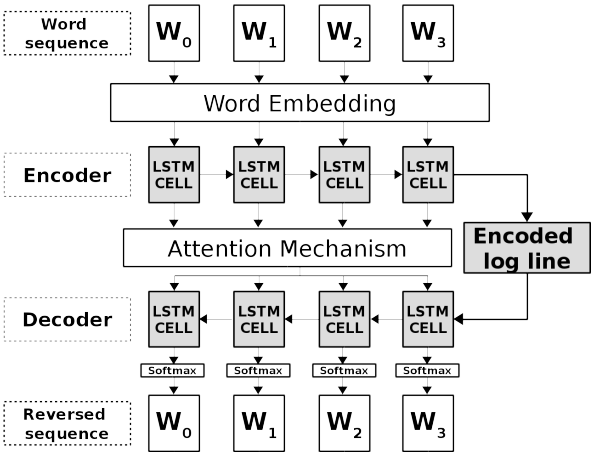


Figure 2. Architecture of our model. We use an attentive RNN auto-encoder to encode log lines.

Instead of using our models for translation, we use it to predict the reverse sequence of our input sequence. We do so because we want the model to learn an embedding space for our log lines and not to translate sentences. Also, in contrast to (Dzmitry Bahdana et al., 2014), we use a single- instead of a bi-directional LSTM (Hochreiter & Schmidhuber, 1997) for encoding our input sequences.

2.2. Learning objective

The learning objective of our problem is, given a sequence of words, correctly predict the reverse sequence, word by word. We calculate the loss of a mispredicted word by using sampled softmax loss (Jean et al., 2014). Sampled softmax loss approximates the categorical cross-entropy loss between the embedded target word and the predicted word. We motivate our choice for using sampled softmax mainly because we assume potentially very large vocabularies in large log files due to the variable parts of the logs.

The learning objective "forces" the model to learn which information is important for reconstructing a log line. In other words, we learn a lossy compression function of our log lines.

2.3. Optimization procedure

To train our model, we use Adam, which is a form of stochastic gradient descent (Kingma & Ba, 2014). During training, we use dropout to prevent overfitting (Srivastava et al., 2014), and gradient clipping to prevent exploding gradients (Pascanu et al., 2013).

3. Experiment setup

For our experiment, we trained the model that we introduced in Section 2. The RNN encoder, the RNN decoder, and the alignment model have 128 units each. The gradients are calculated in mini-batches of 10 log lines and gradients are clipped at 0.5. We trained each model with a learning rate of 0.001 and dropout rate of 0.3. We drew 500 samples for our sampled softmax learning objective. We determined the hyperparameters empirically using a random search strategy.

We pad input- and output sequences that are of a different length with zeros at the end of the sequences. Additionally, we add a special token that marks the beginning and the end of a sequence of words.

We then hierarchically cluster the encoded log lines by using the Farthest Point Algorithm. We use the Euclidean distance as a distance metric and a clustering threshold of 0.50, which we empirically determined.

We compare our approach to two state-of-the-art signature extraction algorithms, LogCluster (Vaarandi & Pihelgas, 2015) and IPLoM (Makanju et al., 2012). We chose these two approaches for two reasons. First, they scored high in a side-by-side comparison (He et al., 2016). Second, both approaches are capable of finding log clusters when the number of log clusters is not specified upfront. We used the author's implementation of LogCluster for our experiments[1], and the implementation provided by He at al. (He et al., 2016)[2]. IPLoM supports four hyperparameters, a Cluster goodness threshold, a partition support threshold and an upper and lower bound for clustering. The best performing hyper parameters of IPLoM were a Cluster goodness

[1] http://ristov.github.io/logcluster/logcluster-0.08.tar.gz

[2] https://github.com/cuhk-cse/logparser/tree/20882dabb01aa6e1e7241d4ff239121ec978a2fe

threshold of 0.125, a partition support threshold of 0.0, a lower bound of 0.25 and an upper bound of 0.9.

LogCluster has two hyperparameters: 'support', the minimum number of supported log lines to cluster two loglines and 'wfreq', the word frequency required to substitute a word with a placeholder. We ran the LogCluster algorithm with a 'support'-value of 2 and 'wfreq'-value of 0.9. For some log lines, LogCluster did not extract a matching signature. We assigned each of such log lines to their own cluster.

3.1. Evaluation Metrics

To evaluate the quality of the extracted signatures, we evaluate the clusters that we have found. We use two metrics to evaluate the quality of our clusters, the Silhouette score and the V-Measure.

The Silhouette score measures the relative quality of clusters (Rousseeuw, 1987) by calculating and relating intra- and inter-cluster distances. The Silhouette score ranges from -1.0 to 1.0. A score of -1.0 indicates many overlapping clusters and a score of 1.0 means perfect clusters.

The V-Measure is a harmonized mean between the homogeneity and completeness of clusters and captures a clustering solution's success in including all and only data-points from a given class in a given cluster (Rosenberg & Hirschberg, 2007). The V-Measure addresses the clustering "matching" problem and is in this context more appropriate than the F1-score.

3.2. Datasets

To test our idea, we created a forensic log. We extracted this forensic log from a Linux system disk image using log2timeline[3]. Log2timeline is a forensic tool that extracts event information from storage media and combines this information in a single file. This log dataset consists of 11023 log messages and 856 signatures. We manually extracted the signatures of these logs and verified their correctness using the Linux source code. The vocabulary size, i.e. the number of unique words, of our dataset is 4282 and the maximum log message length is 135. Due to the nature of forensic logs, we assume that the number of unique words will grow when in larger logs.

4. Results and discussion

In this section, we present the results of our experiments and discuss our findings. Table 1 summarizes the results. The columns, from left to right contain the approach, the Silhouette score of the found clusters and the V-measure of found clusters.

Table 1. Experiment results.

Approach	Silh.	V-Meas.
(Vaarandi & Pihelgas, 2015)	N/A	0.881
(Makanju et al., 2012)	N/A	0.824
RNN-AE + Cluster (ours)	0.749	**0.944**

LogCluster creates log line clusters that have a homogeneity of 1.00 and completeness of 0.777, which yields a V-measure of 0.881. IPLoM creates log line clusters that have a homogeneity of 0.761 and a completeness of 0.898, which yields a V-measure of 0.824. Our approach creates log line clusters that have a homogeneity of 0.990 and a completeness of 0.905, which yields a V-measure of 0.944. Additionally, the clusters formed by our approach have a Silhouette score of 0.749.

None of the tested approaches manages to find all signatures of our forensic log perfectly. In contrast to that, in He et al.'s evaluation (He et al., 2016), both LogCluster and IPLoM to perfect F1 score. We explain this difference by the fact that our dataset is more difficult to analyze than the datasets presented in He et al.'s evaluation. The most difficult dataset had 376 signatures, but 4.7 million log lines, whereas our dataset has only 11023 log lines and 856 signatures. Both IPLoM and LogCluster have been designed the first case with few signatures and many log lines per signature.

Our approach creates almost homogeneous clusters. We illustrate the clusters that are found in Figure 3. Figure 3 shows a sample of 15 log lines and how they are hierarchically clustered together. When two log lines are identical, they have a distance close to zero, which means they are embedded almost on the same spot. If they are related, they are closer to each other than other log lines. Our approach makes very few mistakes of grouping together log lines that do not belong together. Instead, the clusters are incomplete, which means that some clusters should be grouped together but are not.

One drawback of our approach is the increased computing requirements. IPLoM processes our forensic log in average under a second, LogCluster needs about 23 seconds process it whereas training our model for the presented task needs 715 seconds. This increased performance requirements may become a problem on larger datasets. As with the state-of-the-art algo-

[3]https://github.com/log2timeline/plaso/wiki/Using-log2timeline

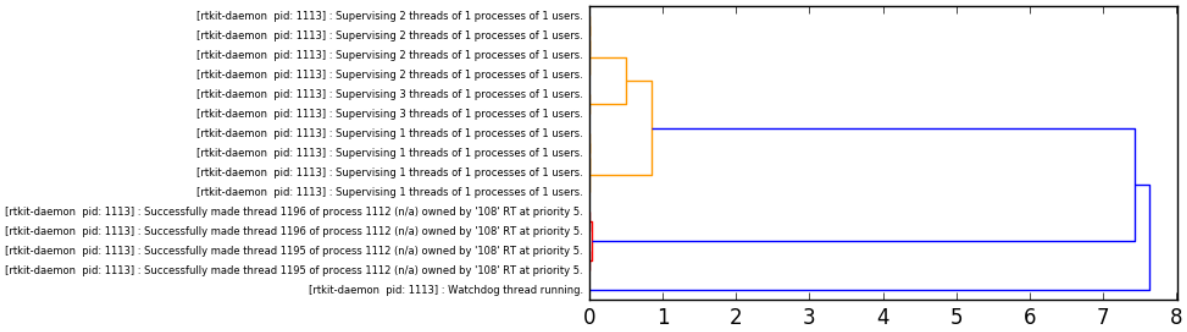


Figure 3. Cluster dendrogram of 15 log lines. The y-axis displays the log lines and x-axis the distance to each other.

rithms, the threshold to separate the clusters has to be manually determined. Our goal is to extract human-understandable log signatures. Currently, we only obtain the centroids of embedded clusters, which allow us to cluster log lines according to their signature. However, these centroids can not effectively be interpreted by humans.

5. Related Work

Log signature extraction has been studied to achieve a variety of goals such as anomaly and fault detection in logs (Vaarandi, 2003; Fu et al., 2009), pattern detection (Vaarandi, 2003; Makanju et al., 2009; Aharon et al., 2009), profile building (Vaarandi & Pihelgas, 2015), forensic analysis (Thaler et al., 2017) or compression of logs (Tang et al., 2011; Makanju et al., 2009). Most of these approaches motivated their signature extraction approach by the large and rapidly increasing volume of log data that needed to be analyzed (Vaarandi, 2003; Makanju et al., 2009; Fu et al., 2009; Aharon et al., 2009; Tang et al., 2011; Vaarandi & Pihelgas, 2015).

Many NLP-related problems have been addressed using neural networks. Collobert et al. were one of the first to successfully apply neural models to a broad variety of NLP-related tasks (Weston & Karlen, 2011). Their approach has been followed by other neural models for similar tasks, e.g. (Dyer et al., 2015; Lample et al., 2016). Also, a variety of language modeling tasks have been tackled using neural architectures, e.g. (Dzmitry Bahdana et al., 2014; Cho et al., 2014; Sutskever et al., 2014).

Auto-encoders have been successfully applied to clustering tasks. For example, auto-encoders have been used to cluster text and images (Xie et al., 2015) or variational recurrent auto-encoders have been used to cluster music snippets (Fabius & van Amersfoort, 2014).

6. Conclusions and future work

We have presented an approach to use an attentive RNN auto-encoder models to address the problem of extracting signatures from forensic logs. We use the auto-encoder to learn a representation of our forensic logs, and cluster the embedded logs to retrieve our signatures. This approach finds signature clusters in our forensic log dataset with a V-Measure of 0.94 and a Silhouette score of 0.75. These results are comparable to the state-of-the-art approaches.

We plan to extend our work in several ways. So far we have only clustered log lines. To complete our objective, we also need a method for extracting human-readable descriptions of these signature clusters. We plan to use the outputs of the attention network to aid the extraction of log signatures from the clustered log lines. Furthermore, we intend to explore regularization techniques that help improve the quality of the extracted signatures. Finally, we intend to demonstrate the feasibility and competitiveness of our approach on large datasets and datasets with fewer signatures.

Acknowledgments

This work has been partially funded by the Dutch national program COMMIT under the Big Data Veracity project.

References

Aharon, M., Barash, G., Cohen, I., & Mordechai, E. (2009). One graph is worth a thousand logs: Uncovering hidden structures in massive system event logs. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 227–243).

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., & Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*.

Dzmitry Bahdana, Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation By Jointly Learning To Align and Translate. *Iclr 2015*, 1–15.

Fabius, O., & van Amersfoort, J. R. (2014). Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*.

Fu, Q., Lou, J.-g., Wang, Y., & Li, J. (2009). Execution Anomaly Detection in Distributed Systems through Unstructured Log Analysis. *ICDM* (pp. 149–158).

He, P., Zhu, J., He, S., Li, J., & Lyu, M. R. (2016). An evaluation study on log parsing and its use in log mining. *Proceedings - 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2016*, 654–661.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*, 1735–1780.

Jean, S., Cho, K., Memisevic, R., & Bengio, Y. (2014). On Using Very Large Target Vocabulary for Neural Machine Translation.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., & others (2016). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *arXiv preprint arXiv:1611.04558*.

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Makanju, A., Zincir-Heywood, A. N., & Milios, E. E. (2012). A Lightweight Algorithm for Message Type Extraction in System Application Logs. *IEEE Transactions on Knowledge and Data Engineering*, *24*, 1921–1936.

Makanju, A. A. O., Zincir-Heywood, A. N., & Milios, E. E. (2009). Clustering event logs using iterative partitioning. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09* (p. 1255).

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *ICML (3)*, *28*, 1310–1318.

Rosenberg, A., & Hirschberg, J. (2007). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. *EMNLP-CoNLL* (pp. 410–420).

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems* (pp. 3104–3112).

Tang, L., Li, T., & Perng, C.-s. (2011). LogSig : Generating System Events from Raw Textual Logs. *Cikm* (pp. 785–794).

Thaler, S., Menkovski, V., & Petković, M. (2017). Towards a neural language model for signature extraction from forensic logs. *2017 5th International Symposium on Digital Forensic and Security (ISDFS)* (pp. 1–6). IEEE.

Vaarandi, R. (2003). A Data Clustering Algorithm for Mining Patterns From Event Logs. *Computer Engineering* (pp. 119–126).

Vaarandi, R., & Pihelgas, M. (2015). LogCluster - A Data Clustering and Pattern Mining Algorithm for Event Logs. *12th International Conference on Network and Service Management - CNSM 2015* (pp. 1–8).

Weston, J., & Karlen, M. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, *12*, 2493–2537.

Xie, J., Girshick, R., & Farhadi, A. (2015). Unsupervised Deep Embedding for Clustering Analysis. *arXiv preprint arXiv:1511.06335*.

Deep Learning Track

Extended Abstracts

Improving Variational Auto-Encoders using convex combination linear Inverse Autoregressive Flow

Jakub M. Tomczak J.M.Tomczak@uva.nl
Max Welling M.Welling@uva.nl
University of Amsterdam, the Netherlands

Keywords: Variational Inference, Deep Learning, Normalizing Flow, Generative Modelling

Abstract

In this paper, we propose a new volume-preserving flow and show that it performs similarly to the linear general normalizing flow. The idea is to enrich a linear Inverse Autoregressive Flow by introducing multiple lower-triangular matrices with ones on the diagonal and combining them using a convex combination. In the experimental studies on MNIST and Histopathology data we show that the proposed approach outperforms other volume-preserving flows and is competitive with current state-of-the-art linear normalizing flow.

1. Variational Auto-Encoders and Normalizing Flows

Let $\mathbf{x}$ be a vector of D observable variables, $\mathbf{z} \in \mathbb{R}^M$ a vector of stochastic latent variables and let $p(\mathbf{x}, \mathbf{z})$ be a parametric model of the joint distribution. Given data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ we typically aim at maximizing the marginal log-likelihood, $\ln p(\mathbf{X}) = \sum_{i=1}^{N} \ln p(\mathbf{x}_i)$, with respect to parameters. However, when the model is parameterized by a neural network (NN), the optimization could be difficult due to the intractability of the marginal likelihood. A possible manner of overcoming this issue is to apply *variational inference* and optimize the following lower bound:

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\ln p(\mathbf{x}|\mathbf{z})] - \mathrm{KL}\big(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\big), \quad (1)$$

where $q(\mathbf{z}|\mathbf{x})$ is the *inference model* (an *encoder*), $p(\mathbf{x}|\mathbf{z})$ is called a *decoder* and $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$ is the *prior*. There are various ways of optimizing this lower bound but for continuous $\mathbf{z}$ this could be done efficiently through a re-parameterization of $q(\mathbf{z}|\mathbf{x})$ (Kingma & Welling, 2013), (Rezende et al., 2014), which yields a *variational auto-encoder* architecture (VAE).

Typically, a diagonal covariance matrix of the encoder is assumed, *i.e.*, $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z}|\boldsymbol{\mu}(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2(\mathbf{x}))\big)$, where $\boldsymbol{\mu}(\mathbf{x})$ and $\boldsymbol{\sigma}^2(\mathbf{x})$ are parameterized by the NN. However, this assumption can be insufficient and not flexible enough to match the true posterior.

A manner of enriching the variational posterior is to apply a *normalizing flow* (Tabak & Turner, 2013), (Tabak & Vanden-Eijnden, 2010). A (finite) normalizing flow is a powerful framework for building flexible posterior distribution by starting with an initial random variable with a simple distribution for generating $\mathbf{z}^{(0)}$ and then applying a series of invertible transformations $\mathbf{f}^{(t)}$, for $t = 1, \ldots, T$. As a result, the last iteration gives a random variable $\mathbf{z}^{(T)}$ that has a more flexible distribution. Once we choose transformations $\mathbf{f}^{(t)}$ for which the Jacobian-determinant can be computed, we aim at optimizing the following lower bound (Rezende & Mohamed, 2015) :

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}^{(0)}|\mathbf{x})}\Big[\ln p(\mathbf{x}|\mathbf{z}^{(T)}) + \sum_{t=1}^{T} \ln \Big|\det \frac{\partial \mathbf{f}^{(t)}}{\partial \mathbf{z}^{(t-1)}}\Big|\Big] - \mathrm{KL}\big(q(\mathbf{z}^{(0)}|\mathbf{x})||p(\mathbf{z}^{(T)})\big). \quad (2)$$

The fashion the Jacobian-determinant is handled determines whether we deal with *general normalizing flows* or *volume-preserving flows*. The general normalizing flows aim at formulating the flow for which the Jacobian-determinant is relatively easy to compute. On the contrary, the volume-preserving flows design series of transformations such that the Jacobian-determinant equals 1 while still it allows to obtain flexible posterior distributions.

In this paper, we propose a new volume-preserving

Appearing in *Proceedings of Benelearn 2017*.

flow and show that it performs similarly to the linear general normalizing flow.

2. New Volume-Preserving Flow

In general, we can obtain more flexible variational posterior if we model a full-covariance matrix using a linear transformation, namely, $\mathbf{z}^{(1)} = \mathbf{L}\mathbf{z}^{(0)}$. However, in order to take advantage of the volume-preserving flow, the Jacobian-determinant of $\mathbf{L}$ must be 1. This could be accomplished in different ways, *e.g.*, $\mathbf{L}$ is orthogonal matrix or it is the lower-triangular matrix with ones on the diagonal. The former idea was employed by the Hauseholder flow (HF) (Tomczak & Welling, 2016) and the latter one by the linear Inverse Autoregressive Flow (LinIAF) (Kingma et al., 2016). In both cases, the encoder outputs an additional set of variables that are further used to calculate $\mathbf{L}$. In the case of the LinIAF, the lower triangular matrix with ones on the diagonal is given by the NN explicitly.

However, in the LinIAF a single matrix $\mathbf{L}$ could not fully represent variations in data. In order to alleviate this issue we propose to consider K such matrices, $\{\mathbf{L}_1(\mathbf{x}), \ldots, \mathbf{L}_K(\mathbf{x})\}$. Further, to obtain the volume-preserving flow, we propose to use a convex combination of these matrices $\sum_{k=1}^{K} y_k(\mathbf{x})\mathbf{L}_k(\mathbf{x})$, where $\mathbf{y}(\mathbf{x}) = [y_1(\mathbf{x}), \ldots, y_K(\mathbf{x})]^\top$ is calculated using the softmax function, namely, $\mathbf{y}(\mathbf{x}) = \mathrm{softmax}(\mathrm{NN}(\mathbf{x}))$, where $\mathrm{NN}(\mathbf{x})$ is the neural network used in the encoder.

Eventually, we have the following linear transformation with the convex combination of the lower-triangular matrices with ones on the diagonal:

$$\mathbf{z}^{(1)} = \Big(\sum_{k=1}^{K} y_k(\mathbf{x})\mathbf{L}_k(\mathbf{x}) \Big) \mathbf{z}^{(0)}. \tag{3}$$

The convex combination of lower-triangular matrices with ones on the diagonal results again in the lower-triangular matrix with ones on the diagonal, thus, $|\det \Big(\sum_{k=1}^{K} y_k(\mathbf{x})\mathbf{L}_k(\mathbf{x}) \Big)| = 1$. This formulates the volume-preserving flow we refer to as *convex combination linear IAF* (ccLinIAF).

3. Experiments

Datasets In the experiments we use two datasets: the MNIST dataset[1] (LeCun et al., 1998) and the Histopathology dataset (Tomczak & Welling, 2016). The first dataset contains 28×28 images of handwritten digits (50,000 training images, 10,000 validation images and 10,000 test images) and the second one contains 28×28 gray-scaled image patches of histopathology scans (6,800 training images, 2,000 validation images and 2,000 test images). For both datasets we used a separate validation set for hyper-parameters tuning.

[1]We used the dynamically binarized dataset as in (Salakhutdinov & Murray, 2008).

Table 1. Comparison of the lower bound of marginal log-likelihood measured in nats of the digits in the MNIST test set. Lower value is better. Some results are presented after: ♣ (Rezende & Mohamed, 2015), ♢ (Dinh et al., 2014), ♠ (Salimans et al., 2015), ♡ (Tomczak & Welling, 2016)

METHOD	$\leq \ln p(\mathbf{x})$
VAE	−93.9
VAE+NF (T=10) ♣	−87.5
VAE+NF (T=80) ♣	**−85.1**
VAE+NICE (T=10) ♢	−88.6
VAE+NICE (T=80) ♢	−87.2
VAE+HVI (T=1) ♠	−91.7
VAE+HVI (T=8) ♠	−88.3
VAE+HF(T=1) ♡	−87.8
VAE+HF(T=10) ♡	−87.7
VAE+LINIAF	−85.8
VAE+CCLINIAF(K=5)	**−85.3**

Set-up In both experiments we trained the VAE with 40 stochastic hidden units, and the encoder and the decoder were parameterized with two-layered neural networks (300 hidden units per layer) and the gate activation function (van den Oord et al., 2016), (Tomczak & Welling, 2016). The number of combined matrices was determined using the validation set and taking more than 5 matrices resulted in no performance improvement. For training we utilized ADAM (Kingma & Ba, 2014) with the mini-batch size equal 100 and one example for estimating the expected value. The learning rate was set according to the validation set. The maximum number of epochs was 5000 and early-stopping with a look-ahead of 100 epochs was applied. We used the *warm-up* (Bowman et al., 2015), (Sønderby et al., 2016) for first 200 epochs. We initialized weights according to (Glorot & Bengio, 2010).

We compared our approach to linear normalizing flow (VAE+NF) (Rezende & Mohamed, 2015), and finite volume-preserving flows: NICE (VAE+NICE) (Dinh et al., 2014), HVI (VAE+HVI) (Salimans et al., 2015), HF (VAE+HF) (Tomczak & Welling, 2016), linear IAF (VAE+LinIAF) (Kingma et al., 2016) on the MNIST data, and to VAE+HF on the Histopathology data. The methods were compared according to the lower bound of marginal log-likelihood measured on the test set.

Discussion The results presented in Table 1 and 2 for MNIST and Histopathology data, respectively,

Table 2. Comparison of the lower bound of marginal log-likelihood measured in nats of the image patches in the Histopathology test set. Higher value is better. The experiment was repeated 3 times. The results for VAE+HF are taken from: ♡ (Tomczak & Welling, 2016).

METHOD	$\leq \ln p(\mathbf{x})$
VAE ♡	1371.4 ± 32.1
VAE+HF (T=1) ♡	1388.0 ± 22.1
VAE+HF (T=10) ♡	1397.0 ± 15.2
VAE+HF (T=20) ♡	1398.3 ± 8.1
VAE+LINIAF	1388.6 ± 71
VAE+CCLINIAF(K=5)	$\mathbf{1413.8 \pm 22.9}$

reveal that the proposed flow outperforms all volume-preserving flows and performs similarly to the linear normalizing flow with large number of transformations. The advantage of using several matrices instead of one is especially apparent on the Histopathology data where the VAE+ccLinIAF performed better by about 15nats than the VAE+LinIAF. Hence, the convex combination of the lower-triangular matrices with ones on the diagonal seems to allow to better reflect the data with small additional computational burden.

Acknowledgments

The research conducted by Jakub M. Tomczak was funded by the European Commission within the Marie Skłodowska-Curie Individual Fellowship (Grant No. 702666, "Deep learning and Bayesian inference for medical imaging").

References

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Burda, Y., Grosse, R., & Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

Dinh, L., Krueger, D., & Bengio, Y. (2014). Nice: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *AISTATS* (pp. 249–256).

Householder, A. S. (1958). Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)*, *5*, 339–342.

Kingma, D., & Ba, J. (2014). ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., Salimans, T., Józefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improving variational inference with inverse autoregressive flow. *NIPS*.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

LeCun, Y., Cortes, C., & Burges, C. J. (1998). The MNIST database of handwritten digits.

Li, Y., & Turner, R. E. (2016). Rényi divergence variational inference. *arXiv preprint arXiv:1602.02311*.

Oord, A. v. d., Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *ICML*, 1747–1756.

Rezende, D., & Mohamed, S. (2015). Variational Inference with Normalizing Flows. *ICML* (pp. 1530–1538).

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Salakhutdinov, R., & Murray, I. (2008). On the quantitative analysis of deep belief networks. *ICML* (pp. 872–879).

Salimans, T., Kingma, D. P., & Welling, M. (2015). Markov chain Monte Carlo and Variational Inference: Bridging the gap. *ICML* (pp. 1218–1226).

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., & Winther, O. (2016). Ladder variational autoencoders. *arXiv preprint arXiv:1602.02282*.

Tabak, E., & Turner, C. V. (2013). A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, *66*, 145–164.

Tabak, E. G., & Vanden-Eijnden, E. (2010). Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, *8*, 217–233.

Tomczak, J. M., & Welling, M. (2016). Improving Variational Auto-Encoders using Householder Flow. *arXiv preprint arXiv:1611.09630*.

van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., & Kavukcuoglu, K. (2016). Conditional image generation with pixelcnn decoders. *Advances in Neural Information Processing Systems* (pp. 4790–4798).

The use of shallow convolutional neural networks in predicting promotor strength in *Escherichia coli*

Jim Clauwaert[1] JIM.CLAUWAERT@UGENT.BE
Michiel Stock[1] MICHIEL.STOCK@UGENT.BE
Marjan De Mey[2] MARJAN.DEMEY@UGENT.BE
Willem Waegeman[1] WILLEM.WAEGEMAN@UGENT.BE

[1]KERMIT, Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Coupure links 653, 9000, Ghent, Belgium
[2]InBio, Centre for Industrial Biotechnology and Biocatalysis, Ghent University, Coupure links 653, 9000, Ghent, Belgium

Keywords: artificial neural networks, promoter engineering, E. coli,

Abstract

Gene expression is an important factor in many processes of synthetic biology. The use of well-characterized promoter libraries makes it possible to obtain reliable estimates on the transcription rates in genetic circuits. Yet, the relation between promoter sequence and transcription rate is largely undiscovered. Through the use of shallow convolutional neural networks, we were able to create models with good predictive power for promoter strength in *E. coli*.

1. Introduction

The binding region of the transcription unit, called the promoter region, is known to play a key role in the transcription rate of downstream genes. In Eubacteria, the sigma factor (σ) binds with the RNA polymerase subunit ($\alpha\beta\beta'\omega$) to create RNA polymerase. Being part of the RNA polymerase holoenzyme, the sigma element acts as the connection between RNA polymerase and DNA (Gruber & Gross, 2003). The use of prokaryotic organisms such as *E. coli* are indispensable in research and biotechnological industry. As such, multiple studies have investigated creating predictive models for promoter strength (De Mey, 2007; Meng et al., 2017). As of now, existing models are trained using small in-house promoter libraries, without evaluation on external public datasets.

Preliminary work. Under review for Benelearn 2017. Do not distribute.

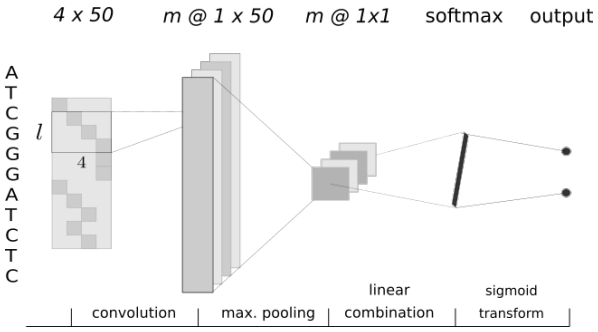


Figure 1. Basic layout of the first model. The sequence, transformed into a 4×50 binary image, is evaluated by the first convolutional layer outputting a vector of scores for every $4 \times l$ kernel (m). Rectified outputs are maximum pooled and fed into the third layer. The sigmoid transform of the softmax layer results in a probability score for each class.

Following the recent success of artificial neural networks (Alipanahi et al., 2015), inspiration was taken to create specialized models for promoter strength. Due to the low amount of promoter libraries, models were instead trained to predict the presence of a promoter region within a DNA sequence, using the more abundant ChIP-chip data. To evaluate whether the model gives a good indication of promoter strength, the predicted score of the model was compared with the given promoter strength scores of existing promoter libraries. The use of several custom model architectures are considered.

2. Results

ChIP-chip data of the σ^{70} transcription factor was used from cells in the exponential phase (Cho et al.,

Table 1. Performance measures of the models on the test set (AUC) and external datasets (Spearman's rank correlation). given values are the averaged scores of the repeated experiments. The standard deviation is given within brackets

kernel size	test set 38984 seq.	Anderson 19 prom.	Brewster(2012) 18 prom.	Davis(2011) 10 prom.	Mutalik[a] (2013) 118 prom.	Mutalik[b](2013) 137 prom.
M1 4×25	0.79 (0.02)	0.15 (0.19)	0.81 (0.09)	0.74 (0.12)	0.40 (0.07)	0.22 (0.07)
4×10	0.79 (0.02)	0.25 (0.16)	0.81 (0.11)	0.77 (0.08)	0.45 (0.04)	0.23 (0.04)
M2 4×25	0.78 (0.02)	0.20 (0.12)	0.74 (0.11)	0.81 (0.14)	0.50 (0.08)	0.16 (0.05)
4×10	0.77 (0.02)	-0.16 (0.14)	0.78 (0.07)	0.68 (0.10)	0.41 (0.06)	0.12 (0.07)
M3 4×25	0.79 (0.02)	0.38 (0.14)	0.82 (0.10)	0.84 (0.08)	0.53 (0.07)	0.25 (0.10)
4×10	0.76 (0.01)	0.70 (0.14)	0.83 (0.06)	0.84 (0.08)	0.47 (0.08)	0.41 (0.15)

[a]part of promoter library with variety within the -35 and -10 box
[b]part of promoter library with variation over the whole sequence

2014). ChIP-chip experiments give an affinity measure between the RNA polymerase holoenzyme and DNA, pinpointing possible promoter sites. Due to the high noise of ChIP-chip data, a classification approach was taken to build the model. The convolutional neural network is fed binary images (4×50) of the sequences following the work of Alipanahi (2015). Promoter sequences within the dataset are determined using both a transcription start site mapping (Cho et al., 2014), and through selection of the highest peaks.

Multiple architectures have been considered, with small changes applied to the basic model (M1) given in Figure 1. The general model architecture is largely based upon the work of Alipanahi (2015). To retain positional data of high-scoring subsequences, a second model (M2) uses the pooled outputs of subregions of the sequence based upon the length (l) of the kernel (motif). Thus, a motif with length $l = 10$ creates $50/l = 5$ outputs for every kernel in the convolutional layer. A further adjustment first splits the sequence into subsequences according to the motif length. This reduces the number of outputs created in the first layer of the previous model. When training 4×10 kernels, five subsequences are created. Motifs are trained uniquely on one of the parts. The third model (M3) retains positional information, while having the same complexity as M1, albeit at a cost of flexibility.

To get an insight into the performances of the models, the use of long (4×25) and short (4×10) motifs have been evaluated. model training was repeated 50 times to account for model variety. In this study we trained models for binary classification, predicting the existence of a promoter region within a given DNA sequence. To verify whether given models can furthermore give reliable predictions on promoter strength, following the idea that stronger promoters are more likely to be recognized, promoter libraries have been ranked on the probability scores of the model. The Spearman's rank correlation coefficient is used as a measure of similarity of ranking between the probability scores and given scores within each promoter library. Table 1 gives an overview of the performances on the test set and external datasets.

3. Discussion

We found that the introduction of the proposed model architectures shows significant improvement on ranking known promoter libraries by promoter strength. The results furthermore show that retaining positional data can offer non-trivial boosts to smaller kernel sizes. Yet, the M2 results show that these advantages are outweighed for smaller kernels, where an increased model complexity reduces overall scores. For longer motifs, M2 still offers a boost in performance as the increase in features compared to M1 is small. M3, gaining positional information of the motifs without the cost of any additional complexity shows the best results for each dataset using both short and long motifs. The use of small kernels, with the exception of M2, generally offers better scores. The performance of the model to identify promoter regions on the test set shows little variations, with AUC scores reaching 0.80. Further optimizations to both the architecture of the model and selection of hyperparameters can prove to further increase model performance.

4. Conclusion

A comprehensive tool for promoter strength prediction, in line with the creation of the ribosome binding site calculator (Salis, 2011), is highly anticipated in the research community. This study shows the potential of using an indirect approach in creating predictive models for promoter strength.

References

Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnol*, *33*, 831–838.

Brewster, R. C., Jones, D. L., & Phillips, R. (2012). Tuning Promoter Strength through RNA Polymerase Binding Site Design in Escherichia coli. *PLoS Computational Biology*, *8*.

Cho, B.-K., Kim, D., Knight, E. M., Zengler, K., & Palsson, B. O. (2014). Genome-scale reconstruction of the sigma factor network in Escherichia coli: topology and functional states. *BMC biology*, *12*, 4.

Davis, J. H., Rubin, A. J., & Sauer, R. T. (2011). Design, construction and characterization of a set of insulated bacterial promoters. *Nucleic Acids Research*, *39*, 1131–1141.

De Mey, M. (2007). Construction and model-based analysis of a promoter library for E.coli: An indispensable tool for metabolic engineering. *BMC Biotechnology*, *7*, 34.

Gruber, T. M., & Gross, C. A. (2003). Multiple sigma subunits and the partitioning of bacterial transcription space. *Annual Review of Microbiology*, *57*, 441–66.

Meng, H., Ma, Y., Mai, G., Wang, Y., & Liu, C. (2017). Construction of precise support vector machine based models for predicting promoter strength. *Quantitative Biology*.

Mutalik, V. K., Guimaraes, J. C., Cambray, G., Lam, C., Christoffersen, M. J., Mai, Q.-A., Tran, A. B., Paull, M., Keasling, J. D., Arkin, A. P., & Endy, D. (2013). Precise and reliable gene expression via standard transcription and translation initiation elements. *Nature Methods*, *10*, 354–60.

Salis, H. M. (2011). The ribosome binding site calculator. *Methods in Enzymology*, *498*, 19–42.

Normalisation for painting colourisation

Nanne van Noord NANNE@UVT.NL
Cognitive Science and Artificial Intelligence group, Tilburg University

Keywords: image colourisation, convolutional neural network, normalisation

1. Introduction

Recent work on style transfer (mapping the style from one image onto the content of another) has shown that *Instance Normalisation* (InstanceNorm) when compared to *Batch Normalisation* (BatchNorm) speeds up neural network training and produces better looking results (Ulyanov et al., 2016; Dumoulin et al., 2016; Huang & Belongie, 2017). While the benefits of normalisation for neural network training are fairly well understood (LeCun et al., 2012; Ioffe & Szegedy, 2015), it is unclear why using instance over batch statistics gives such an improvement. Huang and Belongie (2017) propose the intuitive explanation, that for style transfer BatchNorm centers all samples around a single style whereas InstanceNorm performs *style normalisation*.

Motivated by these developments in style transfer I set out to explore whether similar benefits can be found in another style dependant image-to-image translation task, namely *painting colourisation*. Here we consider painting colourisation a variant of image colourisation, focused on performing the colourisation in a manner that matches the painter's style.

In image colourisation we aim to hallucinate colours given a greyscale image. The main challenge with hallucinating colours is that the task is underconstrained; a pixel with a given greyscale value can be assigned a number of different colours. However, if we are able to recognise *what* is depicted in the image, we may be able to suggest a plausible colourisation. Moreover, for paintings if we know *who* painted it, we can further narrow down what is plausible. Due to the importance of style for painting colourisation, we set out to compare whether using instance statistics over batch statistics offers similar improvements for painting colourisation as were observed for style transfer (i.e., better looking and more stylised results).

2. Normalisation techniques

In this section we discuss the three normalisation techniques we compare in this work: (1) Batch Normalisation, (2) Instance Normalisation, and (3) Conditional Instance Normalisation (CondInNorm).

(1) BatchNorm. Given a batch of size T, BatchNorm normalises each channel c of its input $x \in R^{T \times C \times W \times H}$ such that it has zero-mean and unit-variance (Ioffe & Szegedy, 2015). Formally, BatchNorm is defined as:

$$y_{tijk} = \gamma_i \left(\frac{x_{tijk} - \mu_i}{\sigma_i} \right) + \beta_i, \tag{1}$$

where μ_i and σ_i describe the mean and standard deviation for channel C_i across the spatial axes W and H, and the batch of size T Additionally, for each channel BatchNorm keeps track of a pair of learned parameters γ_i and β_i, that scale and shift the normalised value such that they may potentially recover the original activations if needed (Ioffe & Szegedy, 2015).

(2) InstanceNorm. Ulyanov et al. (2016) modify BatchNorm such that μ_i and σ_i are calculated independently for all T instances in the batch. However, γ and β are still shared.

(3) CondInNorm. CondInNorm is an extension of InstanceNorm which conditions the γ and β parameters on the style (Dumoulin et al., 2016). In this case γ and β become $N \times C$ matrices, where N is equal to the number of styles being modelled. In this work we use the painter as a proxy for the style and instead condition on the painter.

3. Method

Recent work has shown that Convolutional Neural Networks (CNN) can obtain sufficient visual understanding to perform automatic image colourisation (Isola et al., 2016). In this work we extend this to

Table 1. Painting colourisation results measured using RMSE across all pixels, and PSNR in RGB space. "Greyscale" is the result of predicting 0 for the **ab** channels.

Method	RMSE	PSNR
Greyscale	0.172	24.88
BN	0.146	23.26
IN	0.149	23.31
CIN	0.145	23.34

painting colourisation by implementing[1] a CNN based on the "U-Net" architecture used in (Isola et al., 2016). A visualisation of the network architecture can be found in Figure 1. The arrows between the layers in the encoder and decoder represent skip connections, which enable a direct mapping between layers at the same spatial scale.

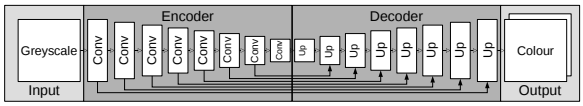


Figure 1. Visualisation of the network architecture. *Conv* refers to a convolutional layer, and *Up* combines upsampling with a convolutional layer. All convolutional layers are followed by a normalisation layer.

Using a CNN we learn a mapping $Y = F(X)$ from the luminance (L) channel of a CIE Lab image $X \in \mathbb{R}^{H \times W}$ to the quantised **ab** colour channels $Y \in \mathbb{R}^{H \times W \times 2 \times Q}$. Where H, W are the image height and width, and Q the number of bins used to quantise the **ab** channels. The predicted histograms across colour bins are converted to colour values by taking the weighted sum of the bins.

4. Experiment

The painting colourisation performance of my CNN using different normalisation methods is evaluated on a subset of the "*Painters by Numbers*" dataset as published on Kaggle[2]. We select the painters who have at least 5 artworks in the dataset, which results in a dataset consisting of 101.580 photographic reproductions of artworks produced by a total of 1.678 painters. All images were rescaled such that the shortest side was 256 pixels, and subsequently a 224×224 crop was extracted for analysis. Table 1 shows the quantitative painting colourisation results. Example colourisations are shown in Figure 2.

[1] `https://github.com/nanne/conditional-colour`
[2] `https://www.kaggle.com/c/painter-by-numbers`

Figure 2. Example painting colourisation results.

5. Conclusion

In this work we used a painting colourisation model capable of producing visually appealing colourisations to compare three normalisation techniques. we conclude that using an instance-based normalisation techniques is beneficial for painting colourisation and that conditioning the shifting and scaling parameters on the painter only leads to minimal improvements.

Acknowledgments

The research is supported by NWO (grant 323.54.004).

References

Dumoulin, V., Shlens, J., Kudlur, M., Brain, G., & View, M. (2016). A learned representation for artistic style. *arXiv preprint*.

Huang, X., & Belongie, S. (2017). Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. *ICLR 2017 Workshop track*.

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ICML* (pp. 448–456). JMLR.

Isola, P., Zhu, J.-y., Zhou, T., & Efros, A. A. (2016). Image-to-Image Translation with Conditional Adversarial Networks. *arXiv preprint*.

LeCun, Y., Bottou, L., Orr, G., & Müller, K. (2012). Efficient backprop. *Neural networks: Tricks of the trade*, 9–48.

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv preprint*.

Predictive Business Process Monitoring with LSTMs

Niek Tax n.tax@tue.nl
Eindhoven University of Technology, The Netherlands

Ilya Verenich, Marcello La Rosa {ilya.verenich,m.larosa}@qut.edu.au
Queensland University of Technology, Australia

Marlon Dumas marlon.dumas@ut.ee
University of Tartu, Estonia

Keywords: deep learning, recurrent neural networks, process mining, business process monitoring

1. Introduction

Predictive business process monitoring techniques are concerned with predicting the evolution of running cases of a business process based on models extracted from historical event logs. A range of such techniques have been proposed for a variety of business process prediction tasks, e.g. predicting the next activity (Becker et al., 2014), predicting the future path (continuation) of a running case (Polato et al., 2016), predicting the remaining cycle time (Rogge-Solti & Weske, 2013), and predicting deadline violations (Metzger et al., 2015). Existing predictive process monitoring approaches are tailor-made for specific prediction tasks and not readily generalizable. Moreover, their relative accuracy varies significantly depending on the input dataset and the point in time when the prediction is made.

Long Short-Term Memory networks (Hochreiter & Schmidhuber, 1997) have been shown to deliver consistently high accuracy in several sequence modeling application domains, e.g. natural language processing and speech recognition. Recently, (Evermann et al., 2016) applied LSTMs specifically to predict the next activity in a case. This paper explores the application of LSTMs for three predictive business process monitoring tasks: (i) the next activity in a running case and its timestamp; (ii) the continuation of a case up to completion; and (iii) the remaining cycle time. The outlined LSTM architectures are empirically compared against tailor-made approaches using four real-life event logs.

2. Next Activity and Time Prediction

We start by predicting the next activity in a case and its timestamp. A log of business process executions consists of sequences (i.e., *traces*) of events, where for each event business task (i.e., *activities*) that was executed and the timestamp is known. Typically, the set of unique business tasks seen in a log is rather small, therefore learned representations (such as (Mikolov et al., 2013)) are unlikely to work well. We transform each event into a feature vector using a one-hot encoding on its activity.

If the last seen event occurred just before closing time of the company, it is likely that the next event of the trace will at earliest take place on the next business day. If this event occurred on a Friday and the company is closed during weekends, it is likely that the next event will take place at earliest on Monday. Therefore, the timestamp of the last seen activity is likely to be useful in predicting the timestamp of the next event. We extract two features representing the time domain: the time since the start of the business day, and the time since the start of the business week.

Figure 1 shows different setups that we explore. First, we explore predicting the next activity and its timestamp with two separate LSTM models. Secondly, we explore predicting them with one joint LSTM model. Thirdly, we explore an architecture with n-shared LSTM layers, followed by m task-specific layers. We use cross entropy loss for the predicted activity and mean absolute error (MAE) loss for the predicted time and train the neural network weights with Adam (Kingma & Ba, 2015).

For time prediction we take as baseline the set, bag, and sequence approach described in (van der Aalst et al., 2011). For activity prediction we take as baselines the LSTM based approach of (Evermann et al., 2016) and the technique of (Breuker et al., 2016). Ta-

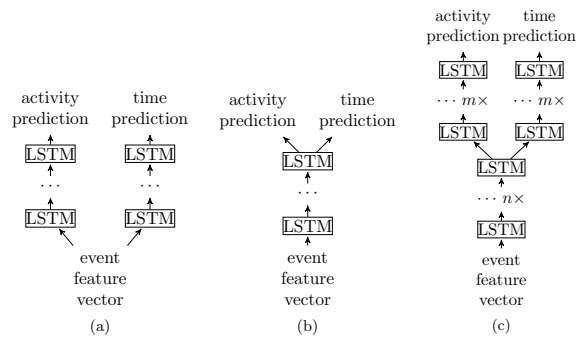


Figure 1. Neural Network architectures with single-task layers *(a)*, with shared multi-tasks layer *(b)*, and with $n+m$ layers of which n are shared *(c)*.

		Helpdesk		BPI'12 W	
Layers	Shared	MAE (time)	Accuracy (act.)	MAE (time)	Accuracy (act.)
3	3	3.77	0.7116	1.58	0.7507
3	2	3.80	0.7118	1.57	0.7512
3	1	3.76	**0.7123**	1.59	0.7525
3	0	3.82	0.6924	1.66	0.7506
2	2	3.81	0.7117	1.58	0.7556
2	1	3.77	0.7119	**1.56**	**0.7600**
2	0	3.86	0.6985	1.60	0.7537
1	1	**3.75**	0.7072	1.57	0.7486
1	0	3.87	0.7110	1.59	0.7431
		Time prediction baselines			
Set		5.83	-	1.97	-
Bag		5.74	-	1.92	-
Sequence		5.67	-	1.91	-
		Activity prediction baselines			
Evermann		-	-	-	0.623
Breuker		-	-	-	0.719

Table 1. Experimental results for the Helpdesk and BPI'12 W logs.

ble 1 shows the MAE of the predicted timestamp of the next event and the accuracy of the predicted activity on two data sets. It shows that LSTMs outperform the baseline techniques, and that architectures with shared layers outperform architectures without shared layers.

3. Suffix Prediction

By repeatedly predicting the next activity, using the method described in Section 2, the trace can be predicted completely until its end. The most recent method to predict an arbitrary number of events ahead is (Polato et al., 2016), which extracts a transition system from the log and then learns a machine learning model for each transition system state. Levenshtein similarity is a frequently used string similarity measure, which is based on the minimal number of insertions, deletions and substitutions needed to transform one string into another. In business processes, activities are frequently performed in parallel, leading to some event in the trace being arbitrarily ordered., therefore we consider it only a minor mistake when two events are predicted in the wrong order. We evaluate suffix predictions with Damerau-Levenshtein similarity, which adds a swapping operation to Levenshtein similarity. Table 2

Method	Helpdesk	BPI'12 W	Env. permit
(Polato et al., 2016)	0.2516	0.0458	0.0260
LSTM	**0.7669**	**0.3533**	**0.1522**

Table 2. Suffix prediction results in terms of Damerau-Levenshtein Similarity.

shows the results of suffix prediction on three data sets. The LSTM outperforms the baseline on all logs.

4. Remaining Cycle Time Prediction

By repeatedly predicting the next activity and its timestamp with the method described in Section 2, the timestamp of the last event of the trace can be predicted, which can be used to predict the remaining cycle time. Figure 2 shows the mean absolute error for each prefix size, for the four logs. As baseline we use the set, bag, and sequence approach described in (van der Aalst et al., 2011), and the approach described in (van Dongen et al., 2008). It can be seen that LSTM consistently outperforms the baselines for the Helpdesk log and the environmental permit log.

5. Conclusions

The foremost contribution of this paper is a technique to predict the next activity of a running case and its timestamp using LSTM neural networks. We showed that this technique outperforms existing baselines on real-life data sets. Additionally, we found that predicting the next activity and its timestamp via a single model (multi-task learning) yields a higher accuracy than predicting them using separate models. We then showed that this basic technique can be generalized to address two other predictive process monitoring problems: predicting the entire continuation of a running case and predicting the remaining cycle time.

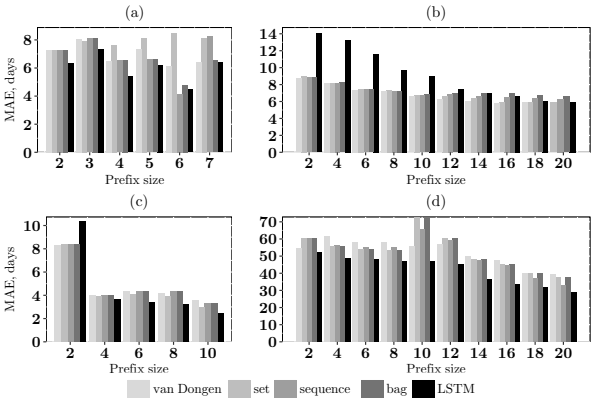


Figure 2. MAE values using prefixes of different lengths for *helpdesk* (a), *BPI'12 W* (b), *BPI'12 W (no duplicates)* (c) and *environmental permit* (d) datasets.

Acknowledgments

This work is accepted and to appear in the proceedings of the International Conference on Advanced Information Systems Engineering (Tax et al., 2017). The source code and supplementary material required to reproduce the experiments reported in this paper can be found at `http://verenich.github.io/ProcessSequencePrediction`. This research is funded by the Australian Research Council (grant DP150103356), the Estonian Research Council (grant IUT20-55) and the RISE_BPM project (H2020 Marie Curie Program, grant 645751).

References

Becker, J., Breuker, D., Delfmann, P., & Matzner, M. (2014). Designing and implementing a framework for event-based predictive modelling of business processes. *Proceedigns of the 6th International Workshop on Enterprise Modelling and Information Systems Architectures* (pp. 71–84). Springer.

Breuker, D., Matzner, M., Delfmann, P., & Becker, J. (2016). Comprehensible predictive models for business processes. *MIS Quarterly*, *40*, 1009–1034.

Evermann, J., Rehse, J.-R., & Fettke, P. (2016). A deep learning approach for predicting process behaviour at runtime. *Proceedings of the 1st International Workshop on Runtime Analysis of Process-Aware Information Systems*. Springer.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference for Learning Representations*.

Metzger, A., Leitner, P., Ivanovic, D., Schmieders, E., Franklin, R., Carro, M., Dustdar, S., & Pohl, K. (2015). Comparing and combining predictive business process monitoring techniques. *IEEE Trans. Systems, Man, and Cybernetics: Systems*, *45*, 276–290.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* (pp. 3111–3119).

Polato, M., Sperduti, A., Burattin, A., & de Leoni, M. (2016). Time and activity sequence prediction of business process instances. *arXiv preprint arXiv:1602.07566*.

Rogge-Solti, A., & Weske, M. (2013). Prediction of remaining service execution time using stochastic Petri nets with arbitrary firing delays. *Proceedings of the International Conference on Service Oriented Computing* (pp. 389–403). Springer.

Tax, N., Verenich, I., La Rosa, M., & Dumas, M. (2017). Predictive business process monitoring with LSTM neural networks. *Proceedings of the International Conference on Advanced Information Systems Engineering* (p. To appear.). Springer.

van der Aalst, W. M. P., Schonenberg, M. H., & Song, M. (2011). Time prediction based on process mining. *Information Systems*, *36*, 450–475.

van Dongen, B. F., Crooy, R. A., & van der Aalst, W. M. P. (2008). Cycle time prediction: when will this case finally be finished? *Proceedings of the International Conference on Cooperative Information Systems* (pp. 319–336). Springer.

Big IoT data mining for real-time energy disaggregation in buildings (extended abstract)

Decebal Constantin Mocanu[⋆] D.C.MOCANU@TUE.NL
Elena Mocanu[⋆] E.MOCANU@TUE.NL
Phuong H. Nguyen[⋆] P.NGUYEN.HONG@TUE.NL
Madeleine Gibescu[⋆] M.GIBESCU@TUE.NL
Antonio Liotta[⋆] A.LIOTTA@TUE.NL
[⋆]Dep. of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands

Keywords: deep learning, factored four-way conditional restricted Boltzmann machines, energy disaggregation, energy prediction

Abstract

In the smart grid context, the identification and prediction of building energy flexibility is a challenging open question. In this paper, we propose a hybrid approach to address this problem. It combines sparse smart meters with deep learning methods, e.g. Factored Four-Way Conditional Restricted Boltzmann Machines (FFW-CRBMs), to accurately predict and identify the energy flexibility of buildings unequipped with smart meters, starting from their aggregated energy values. The proposed approach was validated on a real database, namely the Reference Energy Disaggregation Dataset.

1. Introduction

Unprecedented high volumes of data and information are available in the smart grid context, with the upward growth of the smart metering infrastructure. This recently developed network enables two-way communication between smart grid and individual energy consumers (i.e., the customers), with emerging needs to monitor, predict, schedule, learn and make decisions regarding local energy consumption and production, all in real-time. One possible way to detect building energy flexibility in real-time is by performing energy disaggregation (Zeifman & Roth, 2011). In this paper (Mocanu et al., 2016), we propose an unified framework which incorporates two novel deep learning models, namely Factored Four-Way Conditional Restricted Boltzmann Machines (FFW-CRBM) (Mocanu et al., 2015) and Disjunctive Factored Four-Way Conditional Restricted Boltzmann Machines (DFFW-CRBM) (Mocanu et al., 2017), to perform energy disaggregation, flexibility identification and flexibility prediction simultaneously.

2. The proposed method

Recently, it has been proven that it is possible in an unified framework to perform both, classification and prediction, by using deep learning techniques, such as in (Mocanu et al., 2014; Mocanu et al., 2015; Mocanu et al., 2017). Consequently, in the context of flexibility detection and prediction, we explore the generalization capabilities of Factored Four-Way Conditional Restricted Boltzmann Machines (FFW-CRBM) (Mocanu et al., 2015) and Disjunctive Factored Four-Way Conditional Restricted Boltzmann Machines (DFFW-CRBM) (Mocanu et al., 2017). Both models, FFW-CRBM and DFFW-CRBM, have shown to be successful on outperforming state-of-the-art techniques in both, classification (e.g. Support Vector Machines) and prediction (e.g. Conditional Restricted Boltzmann Machines), on time series classification and prediction in the context of human activity recognition, 3D trajectories estimation and so on. In Figure 1 a high level schematic overview of FFW-CRBM and DFFW-CRBM functionalities is depicted, while for a comprehensive discussion on their mathematical details the interested reader is referred to (Mocanu et al., 2015; Mocanu et al., 2017). The full methodology to perform energy disaggregation can be found in (Mocanu et al., 2016).

Appearing in *Proceedings of Benelearn 2017*.

The full paper has been published in the proceedings of *IEEE International Conference on Systems, Man, and and Cybernetics (SMC 2016)*, Pages 003765-003769, DOI 10.1109/SMC.2016.7844820.

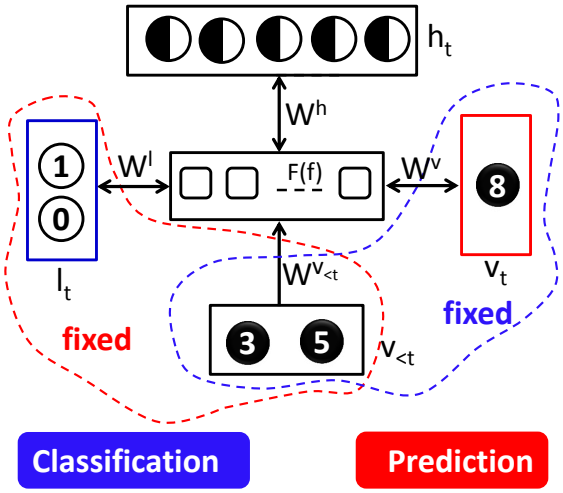


Figure 1. Classification and prediction schemes for FFW-CRBMs (DFFW-CRBM function in a similar manner). To perform classification the value of each neuron from the dotted blue area has to be fixed (i.e. present and history layers) and to let the model to infer the values of the label neurons. To perform prediction the value of each neuron from the dotted red area has to be fixed (i.e. label and history layers) and to let the model to infer the values of the present neurons.

We assessed our proposed framework on the The Reference Energy Disaggregation Dataset (REDD) dataset (Kolter & Johnson, 2011). The results presented in Table 1 and 2 show that both models performed very well obtaining a minimum prediction error on the power consumption of 1.85% and a maximum error of 9.36%, while for the time-of-use prediction the minimum error reached was 1.77% in the case of the electric heater and the maximum error obtained was 8.79% for the refrigerator.

3. Conclusion

In this paper, we proposed a novel IoT framework to perform simultaneously and in real-time flexibility identification and prediction, by making use of Factored Four Way Conditional Restricted Boltzmann Machines and their Disjunctive version. The experimental validation performed on a real-world database shows that both models perform very well, reaching a similar performance with state-of-the-art models on flexibility identification, while having the advantage of being capable to perform also flexibility prediction.

Acknowledgments

This research has been partly funded by the European Union's Horizon 2020 project INTER-IoT (grant number 687283), and by the NL Enterprise Agency under the TKI SG-BEMS project of Dutch Top Sector.

Table 1. Results showing accuracy [%] and balanced accuracy [%] for FFW-CRBM and DFFW-CRBM, when classifying an appliance versus all data.

Appliance	Method	Accuracy [%]	Balanced accuracy [%]
refrigerator	FFW-CRBM	86.23	90.05
	DFFW-CRBM	83.10	91.27
dishwasher	FFW-CRBM	97.42	80.21
	DFFW-CRBM	97.26	87.06
washer dryer	FFW-CRBM	98.83	99.03
	DFFW-CRBM	99.06	92.16
electric heater	FFW-CRBM	99.10	90.58
	DFFW-CRBM	99.03	92.05

Table 2. Results showing the NRMSE [%] obtained to estimate the electrical demand and the time-of-use for four building electrical sub-systems using FFW-CRBM and DFFW-CRBM.

Appliance	Method	Power NRMSE [%]	Time-of-use NRMSE [%]
refrigerator	FFW-CRBM	9.36	8.79
	DFFW-CRBM	9.27	8.71
dishwasher	FFW-CRBM	5.49	5.89
	DFFW-CRBM	5.41	5.87
washer dryer	FFW-CRBM	2.70	2.43
	DFFW-CRBM	2.59	2.44
electric heater	FFW-CRBM	1.86	1.78
	DFFW-CRBM	1.85	1.77

References

Kolter, J. Z., & Johnson, M. J. (2011). REDD: A Public Data Set for Energy Disaggregation Research. *SustKDD Workshop on Data Mining Applications in Sustainability*. San Diego, California, USA.

Mocanu, D. C., Ammar, H. B., Lowet, D., Driessens, K., Liotta, A., Weiss, G., & Tuyls, K. (2015). Factored four way conditional restricted boltzmann machines for activity recognition. *Pattern Recognition Letters*, *66*, 100 – 108.

Mocanu, D. C., Ammar, H. B., Puig, L., Eaton, E., & Liotta, A. (2017). Estimating 3d trajectories from 2d projections via disjunctive factored four-way conditional restricted boltzmann machines. *Pattern Recognition*.

Mocanu, D. C., Mocanu, E., Nguyen, P. H., Gibescu, M., & Liotta, A. (2016). Big iot data mining for real-time energy disaggregation in buildings. *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 003765–003769).

Mocanu, E., Mocanu, D. C., Ammar, H. B., Zivkovic, Z., Liotta, A., & Smirnov, E. (2014). Inexpensive user tracking using boltzmann machines. *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 1–6).

Zeifman, M., & Roth, K. (2011). Nonintrusive appliance load monitoring: Review and outlook. *IEEE Transactions on Consumer Electronics*, *57*, 76–84.

Industry Track

Research Papers

Comparison of Syntactic Parsers on Biomedical Texts

http://wwwen.uni.lu/lcsb

Maria Biryukov MARIA.BIRYUKOV@UNI.LU

Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Belval, Luxembourg

Keywords: syntactic parser, biomedical, text mining

Abstract

Syntactic parsing is an important step in the automated text analysis which aims at information extraction. Quality of the syntactic parsing determines to a large extent the recall and precision of the text mining results. In this paper we evaluate the performance of several popular syntactic parsers in application to the biomedical text mining.

1. Introduction

Biomedical information extraction from text is an active area of research with applications to generation of disease maps, protein-protein interaction networks, drug-disease effects and more. Proper understanding of sentence syntactic structure is a precondition for correct interpretation of its meaning. We used five stay-of-the art syntactic parsers, Stanford, BLLIP and SyntaxNet, in two different experiments. One experiment aimed at evaluation of sentence syntactic complexity; another one tested parsers capability to correctly parse biomedical articles.

1.1. Goals

In this paper we describe an ongoing work towards the following goals:

- Evaluate parser performance on the real-life corpora.
- Identify the most frequent parser errors
- Find good predictors for sentence complexity and parser errors
- Identify working strategies on how to efficiently train parsers for biomedical domain.

1.2. Parsers and models for the comparison

For this study we used five different machine-learning based parsers. Three of these are Stanford parsers based on different linguistic paradigms: PCFG, Factored, and RNN (Manning, 2015). All the three parsers have been trained on an English corpora, which is a mixture of newswire, biomedical texts, English translation of Chinese and Arabic Tree Bank, a set of sentences with technical vocabulary. The fourth parser is the BLLIP parser with two different training models: a) World Street Journal (WSJ) and GigaWord (McClossky, 2015), which is a collection of news and news-like style texts; and b) purely biomedical – Genia (Kim et al., 2003) + Medline. The fifth parser is called Parsey McParseface (Presta & et.al, 2017) and makes part of the Google Labs' TensorFlow (Moshe et al., 2016) deep learning framework. (This parser is refered as "Google" in Table 1). Similar to our experiments with BLLIP, we used Parsey McParseface with various language models: the one provided with the distribution, trained on a multi-domain corpus combining data from OntoNotes, Web Treebank, and updated and corrected Question Treebank. We also trained it on the biomedical gold standard corpora - Genia and CRAFT (see section 3.1 for details about the corpora).

The parsers are briefly introduced bellow:

- PCFG is an accurate unlexicalized parser based on a probabilistic context-free grammar (Klein & Manning, 2003).
- Factored parser combines syntactic structures (PCFG) with semantic dependencies provided by using lexical information (Klein & Manning, 2003). Lexicalization is very helpful when dealing with language ambiguities and helps cor-

Preliminary work. Under review for Benelearn 2017. Do not distribute.

rectly identify dependencies between sentence constituents.

- Recursive neural networks (RNN) parser (Socher et al., 2011) works in two steps: first it uses the parses of PCFG parser to train; then recursive neural networks are trained with semantic word vectors and used to score parse trees. In this way syntactic structure and lexical information are jointly exploited.

- BLLIP is a self-trained parser which is based on a probabilistic generative model with maximal entropy discriminative reranking (McClosky & Charniak, 2008). In the self-training approach, an existing parser parses an unseen data and treats this newly labeled data in combination with the actual labeled data to train a second parser. BLLIP exploits this technique to self-train a parser, initially trained on a standard Penn Treebank corpus, using unlabeled biomedical abstracts.

- Parsey McParseface is an incremental transition-based parser with feature embeddings (Andor et al., 2016) introduced by Google Labs. The parser has appealing characteristics, such as accuracy from 89% on web texts up to about 94% on news and questions, and speed (600 words/second).

2. Biomedical corpus analysis

Biomedical language is known to be difficult for parsers because it is very different from "standard" English. We analyzed a development corpus from the BioNLP 2011 competition (Tsujii et al., 2011) from the point of view of its syntactic and lexical variability. Initially the corpus consisted of 2564 sentences. For this experiment we considered only the sentences with at least one mention of a protein and one predicate (noun, verb or adjective), which could trigger an event. In total we analyzed 875 sentences. Sentence length ranged from 2 to 146 tokens, with an average of 26 tokens.

The underlying assumption for the evaluation of a syntactic complexity was that more complex sentences will result in a larger variability of their parses. We parsed the sentences with BLLIP parser trained on the biomedical model (Genia+MEDLINE).

Suppose that the parser builds a syntactic parse graph G for a given sentence. We are interested in assigning such graph with a score which measures the complexity of the graph (sentence). The syntactic parse graphs very often have a tree-like or a direct acyclic (DAG) structure and thus it makes sense to talk about their depth, i.e. the number of levels in such graph. Let us denote by $Depth(G)$ the depth of the sentence parse graph, and by B_i the number of nodes (tokens) on the i-th level of this graph, i.e. breadth. Then one possible sentence complexity score can be calculated as follows:

$$Score(G) = \sum_{i=1}^{Depth(G)} B_i \cdot i.$$

The more tokens and at the lower depth the higher would be this metric. In Fig. 1 we show how sentence complexity scores are distributed in the corpus. Fig. 2 shows the relation between the sentence length, measured in tokens, and the sentence complexity in our score metric.

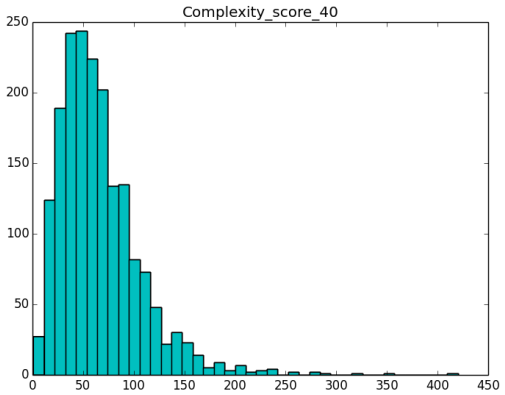


Figure 1. Sentence complexity scores distribution.

For the knowledge extraction pipeline which uses the parser as one of its components it is of interest to assess the quality of the parse and to have an estimate of the probability of a parsing error for a given sentence. For example, correct argument attachment in a sentence can be highly ambiguous and challenging for the parser. BLLIP parser offers an option to output multiple parses for a given sentence. We have picked the top 50 parses per sentence and computed our metric for each parse. The goal was to get an idea on how stable the parse is; the more variation there is in the sentence parses (and thus in their scores), the higher is the chance that the parse may not be correct.

For each sentence we computed 50 parsing scores and then computed the coefficient of variation for each sentence by dividing the standard deviation of the score

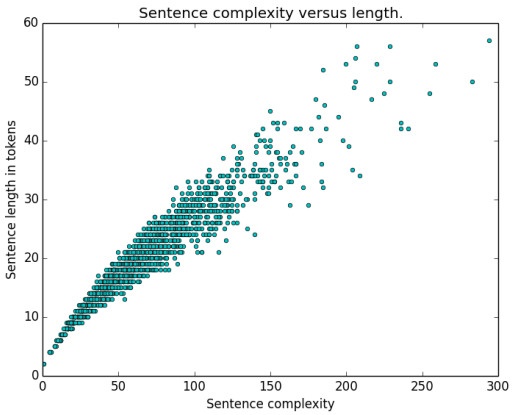


Figure 2. Sentence complexity versus sentence length in tokens.

by the mean of the score:

$$c_v = \frac{\sigma}{\mu}.$$

The distribution of the coefficients of variation c_v of the sentence complexity $Score(G)$ for the BLLIP parser is shown in Fig. 3. As can be seen in this figure

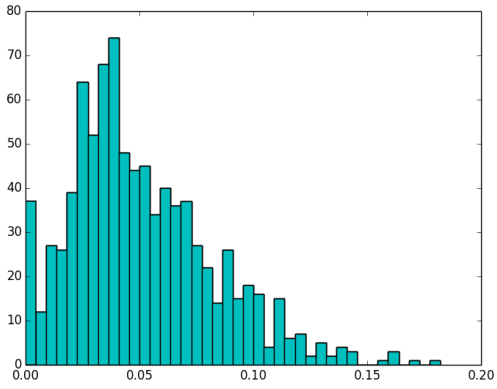


Figure 3. Histogram of the coefficient of variation of the sentence complexity.

the coefficient was in the range: $0 \leq c_v \leq 0.18$. Out of total 850 sentences only 70 sentences had coefficient of variation > 0.10; 341 were in the middle range of $0.05 \leq c_v \leq 0.10$, while 461 were in the lowest variation category ($c_v < 0.5$). Finally for 30 sentences the variation was zero.

These results show that for about 54% of the sentences the BLLIP parse is very stable and only for about $8 - 15\%$ is unstable. This is also reflected by its very good performance in the tests in the following section. The measure c_v for a given sentence can be used as parse error estimator in the event extraction pipeline.

2.1. Sentence content load

What makes biomedical texts so different from "standard" English, are the in-domain words. These are, first of all, names of genes, proteins, molecules, tissues - the list is quite long. We were interested to assess the distribution of complex multi-trigger protein-protein interactions inside the typical biomedical corpus. Given the gold standard annotation of the BioNLP competition data, we calculated distribution of protein names and predicates which could trigger an event in the corpus. It turned out that sentences with only $2 - 3$ mentions of some protein name cover 76% of all the corpus of sentences which contained at least one trigger predicate. The were $11, 4, 2\%$ with four, five and six proteins respectively. This information shows that complex event extraction techniques might not be necessary for protein-protein interaction extraction from the majority of biomedical sentences. It should be noticed however, that text analysis is not limited to only protein-related events.

3. Parser Evaluation for Event Extraction

Semantic relations between individual words and phrases are encoded in the syntactic dependencies. This is the reason why syntactic parsing is an important step towards information extraction. In our applications we aim at finding so-called 'events': functional connections between various biomedical concepts such as proteins, chemicals, diseases, processes. Moreover we are particularly interested in determining the event context: logic and temporal order, coordination, mutual dependency and/or exclusion. Such relations are expressed via abundant use of coordination, prepositions, adverbial and relative clauses. In this paper we evaluate the parsers performance on biomedical texts.

3.1. Test corpus and Data sets

We used three tests corpora. As mentioned in Section 2, for the sentence complexity evaluation we used development corpus of the BioNLP competition (Tsujii et al., 2011). It consisted of 220 documents in which 875 sentences have been parsed by us with BLLIP. To evaluate parsers performance on the biomedical

texts we used two gold standard in-domain corpora. One was released in the framework of Genia project (Kim et al., 2003). It consists of 2000 article abstracts collected with the key-word search "Human, Blood Cells and Transcription Factors". Another corpus is known as "Colorado Richly Annotated Full Text Corpus" (CRAFT) (Verspoor & et.al, 2012). It contains 67 full text articles in a wide variety of biomedical sub-domains. For the evaluation we used about 1000 sentences from each of these corpora. Our choice of test sets size was based on the Genia corpus division into train, development and test sets distributed by D. McClossky (McClosky & Charniak, 2008). As of Genia corpus, we used his division, and created our own for the Craft corpus.

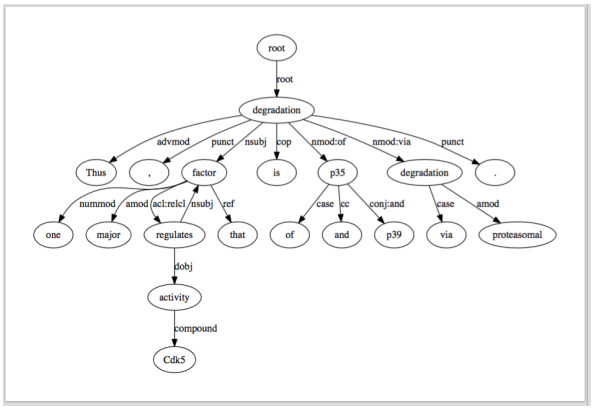


Figure 4. BLLIP parse.

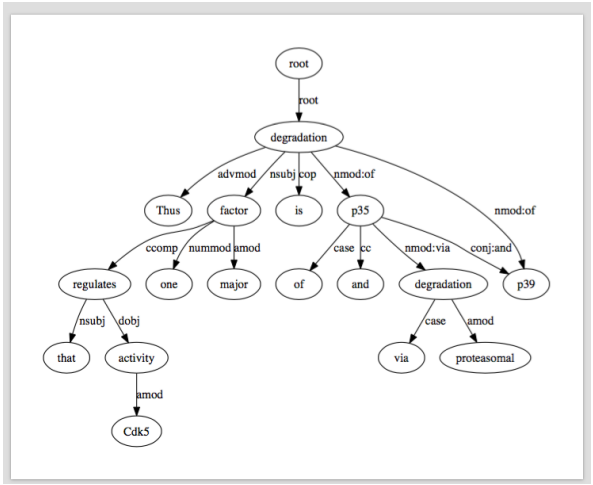


Figure 5. Stanford Factored parse.

3.2. Evaluation and Discussion

Table 1 presents the most important results of each parser. For the overall performance assessment, we adopted evaluation criteria established by the Parser Evaluation challenge (Black et al., 1992), PARSEVAL. These include accuracy of the part of speech tagging, unlabeled attachment score (UAS) which accounts for the correspondence between each node and its parent in the test and gold standard parses, and labeled attachment score (LAS), which, in addition to the parent node correspondence, checks if syntactic relation between two nodes (label on the edge) is the same in the test and gold sets. Given the nature of these two measurements, it is not surprising that LAS are systematically lower than UAS for all the parsers listed in Table 1. However, accuracy of the labeled attachment predefines the extent to which semantic relations between the concepts represented by the nodes would be be correctly interpreted.

It can be seen from the table, that parsers performance depends on how close the test domain is to the training domain. One of the important reasons for that are out-of-domain words. Being unknown to the parser, they present difficulty for the part of speech tagging. The latter is responsible for the assignment of syntactic dependencies between the words. Our analysis shows that part of speech errors are responsible for 30% (for the corresponding training and test domains) to 60% (for different training and test domains) of the errors in dependency assignment. Among all the parsers trained on English corpora, Stanford RNN shows the best result (LAS 0.78) on the Genia corpus. BLLIP trained on Genia + PubMed demonstrates the best performance, followed by Parsey McParseface trained on Genia and CRAFT.

With respect to the biomedical corpora it seems that CRAFT is more difficult than Genia for all but one parsers, either trained on the biomedical texts or not. Detailed corpora investigation is required to answer the question why it is so. However we suppose that certain portions of full texts, such as detailed description of experimental setup, or explanations of the figures, which are not necessarily complete or well formed sentences, contribute to the lower parser scores. Besides, full texts would have larger vocabulary than the abstracts. This fact can be even stronger in our specific training - test setup, due to the sub-domain coverage of both biomedical corpora: Genia, being a narrow-focused one as opposed to a much more diverse Craft. Overall, based on the figures in the Table 1 we think that abstracts are not sufficiently representative for the entire article context to provide an efficient train-

Table 1. Parsers' comparison on test corpus

Parser	Model	Corpus	Test	Pos	UAS	LAS
Stanford	RNN	English	Craft	0.72	0.68	0.63
			Genia	0.83	0.83	**0.78**
BLLIP		Medline+Genia	Craft	0.90	0.76	0.73
			Genia	0.98	0.89	**0.88**
BLLIP		WSJ+GigaWord	Craft	0.77	0.66	0.61
			Genia	0.83	0.74	0.68
Google		English	Genia	0.85	0.57	0.49
Google		Genia	Craft	0.90	0.73	0.64
Google			Genia	0.98	0.90	**0.84**
Google		Craft	Craft	0.97	0.86	0.80
Google			Genia	0.95	0.84	0.78

ing set for the parsers.

In addition to the evaluation presented above one can have a closer look at parsers performance on specific syntactic patterns, such as prepositional attachment or coordination. These constructs carry information about events participants, conditions under which events take place, as well as location at which they happen. At the same time, both, coordination and prepositional attachments, are often difficult to be parsed and attached correctly. Just as an illustration we compare the BLLIP and the Standard Factored parsers on an example of the following sentence: *"Thus, one major factor that regulates Cdk5 activity is degradation of p35 and p39 via proteasomal degradation."*. The graphs of the parses are given in Fig. 4 for BLLIP and Fig. 5 for Stanford Factored respectively. The relevant information that we want to extract in this case is two facts: a) *degradation of both p35 and p39* regulates the *Cdk5 activity*; b) how this degradation happens - via *proteasomal degradation*. The first fact was successfully captured by both parsers, but the mechanism was correctly captured only by BLLIP parser. The Stanford parser failed at prepositional attachment.

Our preliminary evaluation of the parsers performance on specific syntactic patterns shows that the success rate for the prepositional attachment is in the range between 82% to 95%, while coordination is worse, and lies between 66% and 79%.

4. Conclusions

In this paper we have studied five syntactic parsers from three families: Stanford, BLLIP, and Parsey McParseface on biomedical texts. We have seen that the highest performance was reached by BLLIP parser on the Genia test corpus. We have also studied complexity of biomedical sentences in the context of event extraction. We have defined sentence complexity metrics and parse variability metrics which can help to assess parser performance when it is used as part of the knowledge extraction pipeline.

References

Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., & Collins, M. (2016). Globally normalized transition-based neural networks. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers.*

Black, E., Lafferty, J. D., & Roukos, S. (1992). Development and evaluation of a broad-coverage probabilistic grammar of english-language computer manuals. *30th Annual Meeting of the Association for Computational Linguistics, 28 June - 2 July 1992, University of Delaware, Newark, Deleware, USA, Proceedings.* (pp. 185–192).

Kim, J., Ohta, T., Tateisi, Y., & Tsujii, J. (2003). GENIA corpus - a semantically annotated corpus for bio-textmining. *Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology, June 29 - July 3, 2003, Brisbane, Australia* (pp. 180–182).

Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 7-12 July 2003, Sapporo Convention Center, Sapporo, Japan.* (pp. 423–430).

Manning, C. (2015). The Stanford parser: A statistical parser. `https://nlp.stanford.edu/software/lex-parser.shtml`.

McClosky, D., & Charniak, E. (2008). Self-training for Biomedical Parsing. *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA, Short Papers* (pp. 101–104).

McClossky, D. (2015). BLLIP and Models. `https://github.com/BLLIP/bllip-parser/blob/master/MODELS.rst`.

Moshe, L., Marcello, H., & DeLesley, H. (2016). An open-source software library for machine intelligence. `https://www.tensorflow.org`.

Presta, A., & et.al (2017). Syntaxnet: Neural models of syntax. `https://github.com/tensorflow/models/tree/master/syntaxnet`.

Socher, R., Lin, C. C., Ng, A. Y., & Manning, C. D. (2011). Parsing natural scenes and natural language with recursive neural networks. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011* (pp. 129–136).

Tsujii, J., Kim, J.-D., & Pyysalo, S. (2011). Bionlp: 2011. *Proceedings of BioNLP 2011 Workshop*.

Verspoor, K., C. K., & et.al (2012). A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC Bioinformatics*, *13*.

Industry Track

Extended Abstracts

Eskapade: a lightweight, python based, analysis framework

http://eskapade.kave.io

Lodewijk Nauta NAUTA.LODEWIJK@KPMG.NL
KPMG Advisory N.V., Laan van Langerhuize 1, 1186DS, Amstelveen, Netherlands

Max Baak BAAK.MAX@KPMG.NL
KPMG Advisory N.V., Laan van Langerhuize 1, 1186DS, Amstelveen, Netherlands

Keywords: python, data analysis, framework

Abstract

Eskapade is a python framework that accelerates development of advanced analytics workflows in big data environments. The modular set-up allows scalable designs of analysis chains based on commonly available open-source libraries and custom built algorithms using single configuration files with simple syntax: from data ingestion and transformations to ML models including feedback processing.

The framework

Eskapade is a is a python based framework for analytics. The framework employs a modular set up in the entire workflow of Data Science: from data ingestion to transformation to trained model output. Every part of the analysis can be run independently with different parameters from one configuration file with a universal syntax. The modular way of working reduces the complexity of your analysis and makes the reproducibility of the steps undertaken a lot easier.

The framework also includes self-learning software functionality for typical machine learning problems. Combined with easy model evaluation and comparison of predictive algorithms, the life-cycle of an analysis is made simpler. Trained algorithms can predict in real-time or in batch mode and if necessary, trigger the retraining of an algorithm.

Preliminary work. Under review for Benelearn 2017. Do not distribute.

Building an analysis

Eskapade the threshold of making the step from experiments to production when building predictive analytics solutions. The framework can be used to build an analysis, using jupyter for interactive development. Once the analysis is finished you can also use the framework for production purposes by running it in dockers or on your cluster, shortening the time-consuming step of reworking your analysis to production standards.

Since everyone in your team uses the same framework team members can easily exchange code. Moreover, this method of working allows for version control of the analysis.

Running an analysis

Analyses are run from a file called a macro. This macro runs chains and these chains contain links. The links are the fundamental building blocks that do simple operations such as reading data, transforming data, training models and plotting output. Chains can be controlled and rerun individually (for example for retraining) from the command line when you run your macro. In this way it becomes easier to control what is happening at certain points in the analysis, while it is being developed or when it runs in production.

Dependencies

The framework is built on top of a variety of python analytics modules including pandas, scikit-learn, matplotlib, pySpark and pyROOT, and can also be run in jupyter to make the step from experimentation to production easier.

Unsupervised region of interest detection in sewer pipe images: Outlier detection and dimensionality reduction methods

Extended Abstract

Dirk Meijer d.w.j.meijer@liacs.leidenuniv.nl
Arno Knobbe a.j.knobbe@liacs.leidenuniv.nl
LIACS, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands

Keywords: Unsupervised learning, Outlier detection, Dimensionality reduction, Computer Vision

1. Introduction

Sewer pipes require regular inspection to determine the deterioration state and performance, before deciding whether repair or replacement is necessary. Inspections are still mostly performed manually, which leads to subjective and inconsistent ratings for deterioration and urgency, differing between inspectors and even within inspectors (Dirksen et al., 2013).

The SewerSense project aims to investigate the possibilities of automating the analysis of sensory data from sewer pipe inspections and the consequences this would have for the sewer asset management industry.

2. Approach

The currently available data consists mostly of image and video data, grouped by pipe stretch and municipality. Rather than identifying defects in these images directly, we have opted to detect regions of interest (ROIs) in the images and classify these at a later stage.

We make the assumptions that (1) all images are from a forward-facing camera in a sewer pipe, (2) all images are similarly aligned, and (3) the surface of the pipe is similar in appearance for images in a single set. See Figure 1 for an example of what these images may look like. It should be noted with the assumption of "similar appearance" that the concrete and agglomerate often contain a lot of texture, and adjacent pixel values are not necessarily similar.

We define an ROI to be the bounding box of a portion of an image that contains something "unexpected". Note that not all unexpected elements in the sewer pipe will be defects; we also want to detect pipe joints for example. See Figure 2 for an example of the ROIs we hope to detect.

Appearing in *Proceedings of Benelearn 2017*.

Broadly speaking, the pixels in an image can be seen as a feature vector, with the notion that there is some spatial ordering to these features and a high correlation between adjacent pixels. As such, we treat this as an extremely high-dimensional, unsupervised outlier detection problem.

Figure 1. Forward-facing pictures of the same concrete sewer pipe at different locations along the street.

3. Methods

Unsupervised outlier detection methods are analogous to clustering: objects are thought to form clusters in feature space and outliers are those objects that are far away from clusters, or part of small clusters. The model used to fit these clusters must be somewhat restrictive, otherwise the models will overfit on the outliers that are present in the training data.

Since the number of pixels in an image ($\approx 10^6$) is some orders of magnitude greater than the number of images in a set ($\approx 10^3$), some dimensionality reduction is in order before we try to find outliers in the dataset, to ensure our methods don't overfit on the training set. While outlier detection methods for higher dimensionalities exist (Aggarwal & Yu, 2001), these seem to be aimed mostly at sparse data, which our image data is not. Other approaches seem to focus on dimensionality reduction by feature selection and rejection (Zimek et al., 2012), which is not most suited for images.

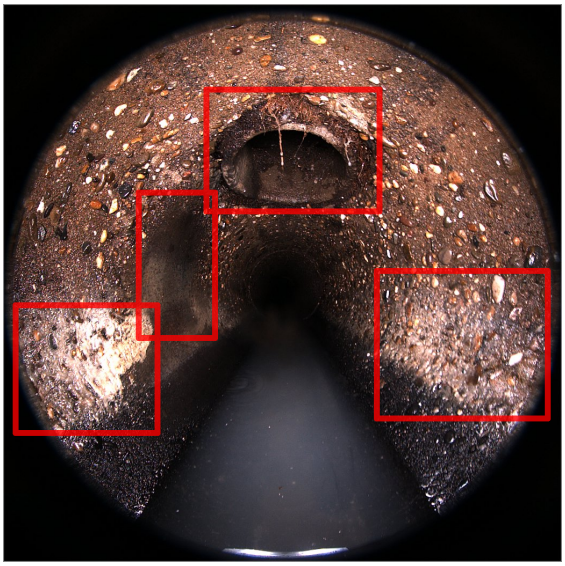


Figure 2. Regions of Interest in a sewer pipe image.

3.1. Principal Component Analysis

Principal component analysis (PCA) projects the d-dimensional data on d orthogonal vectors, in order of decreasing variance present in the data. If we now omit all but the first d_1 vectors, we have a lower dimensionality, while retaining most of the variance (as all omitted vectors have less variance than the retained ones).

The effect PCA has on images is an interesting one: the projection vectors returned by the PCA are shaped like the input image data, and visual inspection shows the collinearity of pixels. We call these *eigenimages* (as the vectors are the eigenvectors of the covariance matrix of the input data), and these have proven to be well suited for image classification tasks such as facial recognition (Turk & Pentland, 1991).

In our application, we can find these eigenimages for an image set and smoothen the image by getting rid of the contribution of all but the first d_1 eigenimages. Every regular occurrence in the image set should be contained in the first few eigenimages, thus present in the smoothed images. By thresholding the difference image of the smoothed image and the original, we should be able to find regions of interest.

There are a few issues with this approach. Firstly, the PCA relies on inverting the covariance matrix of the dataset to find the eigenvectors, which can very quickly become challenging in the case of large sets of large images. Secondly, because of the texture present in the images, we need to have *some* level of spatial invariance, which the PCA does not have.

3.2. Convolutional Autoencoders

An *autoencoder* is a type of artificial neural network, where the target output is identical to the input, and the intermediate layers have fewer neurons than the input and output layers. The result is that the network learns generalization of the input in the compacted layers. When the autoencoder uses strictly linear transfer functions, it is very similar to PCA, estimating a linear mapping onto a lower dimensionality. The interesting application comes of course from using non-linear transfer functions to learn a non-linear mapping.

Convolutional neural networks have been very successful in image classification and bring two new elements to neural networks: convolutional layers and pooling layers. The convolutional layer acts like a filter bank, turning the image into a series of filter responses which contain information about the local structure in the image. Unlike a static filter bank though, these filters are learned from the input data. The pooling layer performs a dimensionality reduction over a neighborhood in the image. This is often max-pooling, taking the maximum value of the filter responses per filter in a specific region. This introduces some spatial invariance, which is well suited for images.

Putting these two approaches together, a convolutional autoencoder can learn a non-linear dimensionality reduction in an unsupervised way, with some spatial invariance that can handle the image texture in a better way than the PCA can. We believe this may prove successful for detecting the ROIs.

4. Anticipated Results

We are working on assembling a labeled dataset. With such a dataset we will investigate and report on the effectiveness of dimensionality reduction methods and outlier detection methods as ROI detection techniques in image sets. We expect the convolutional autoencoder to outperform PCA, and we will look into other possibilities to overcome the issues faced by the PCA that initially make it unsuitable for this task.

Acknowledgments

SewerSense is funded by the NWO/TTW TISCA programme and implemented by researcher at Leiden University and Delft University of Technology. The images used have been provided by vandervalk+degroot.

References

Aggarwal, C. C., & Yu, P. S. (2001). Outlier detection for high dimensional data. *ACM Sigmod Record* (pp. 37–46).

Dirksen, J., Clemens, F., Korving, H., Cherqui, F., Le Gauffre, P., Ertl, T., Plihal, H., Müller, K., & Snaterse, C. (2013). The consistency of visual sewer inspection data. *Structure and Infrastructure Engineering*, *9*, 214–228.

Turk, M. A., & Pentland, A. P. (1991). Face recognition using eigenfaces. *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on* (pp. 586–591).

Zimek, A., Schubert, E., & Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, *5*, 363–387.

Service Revenue Forecasting in Telecommunications: A Data Science Approach

Dejan Radosavljevik D.RADOSAVLJEVIKJ@LIACS.LEIDENUNIV.NL
LIACS, Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands

Peter van der Putten P.W.H.VAN.DER.PUTTEN@LIACS.LEIDENUNIV.NL
LIACS, Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands

Keywords: telecommunications, data science, revenue forecasting

Abstract

This paper discusses a real-world case of revenue forecasting in telecommunications. Apart from the method we developed, we will describe the implementation, which is the key factor of success for data science solutions in a business environment. We will also describe some of the challenges that occurred and our solutions to them. Furthermore, we will explain our unorthodox choice for the error measure. Last, but not least we will present the results of this process.

1. Service revenue forecasting

Given the importance of the process of revenue forecasting, it is surprising there is not a lot of research on this topic. There is literature available regarding the government sector (Fullerton, 1989; Feenberg et al., 1988), airline industry (Harris, 2006) and online firms (Trueman et al., 2001). However, none of them describe this process in mobile telecommunications.

We focus on the postpaid segment of mobile telecommunications, where service revenues can be split into fixed (subscription fee which already contains certain services) and variable (based on additional usage of services not included in the subscription). Furthermore, operators charge differently for using services while abroad (roaming) and for making international calls. Operators also charge other operators interconnect fees for incoming calls (a customer from operator B calls a customer from operator A).

Appearing in *Proceedings of Benelearn 2017*.

The operator where we deployed the research has about two million postpaid customers with more than 4000 combinations of differently priced rate plans and voice, SMS and Internet bundles. In order to forecast the revenue figure for one month, one has to account not only for the different usage patterns throughout the year, but also the inflow of new customers, changes in contract of the existing ones and the loss of customers to competition. The systems that are used for actual customer billing are not built for simulation of revenues, as these are too embedded into operational processes. This makes the task of forecasting revenue across different scenarios far from trivial.

2. Data collection and understanding

Unlike in an academic research setting where seeking the best algorithm is the key to solving the problem and getting data is as easy as downloading a data set, in a business setting this represents a large chunk of the total work. Typically, the data is not available from a single data source, let alone structured in a format suitable for machine learning. The first thing we needed to do is unify the data (invoice data, usage data, interconnect data, rate plans/bundles and inflow and outflow of customers) into a single sandbox which we can use to construct our flat table. We chose to use Teradata (2017) as we had the most administrative flexibility there. Any other database system would do. Also, we used the open source tools R (2008) for moving data between data sources and KNIME (Berthold et al., 2007) for automating the process.

Next, we needed to restructure the data, as it was recorded in a database the same way it is presented to customers on an invoice. However, not all invoice items are service revenues (e.g. the handset fee and application purchases are not service revenues). Non

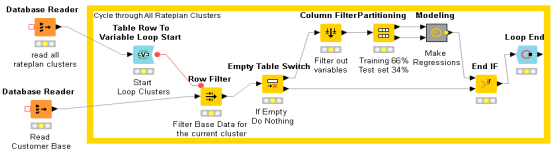


Figure 1. Modeling Workflow in KNIME

service revenues are out of scope. Isolating the service revenues required a lot of expert help as each revenue type has a different code. Importing and fully aligning the data with what is officially billed and reported took almost two months of work. Next to this, distinction between usage that is billed vs usage that is already part of the subscription was necessary. Last, but not least, interconnect usage and usage abroad were added separately as they are billed in a different way.

3. Clustering the data and modeling the service revenues

It is not feasible to come up with a single service revenue model for all customers due to the many different rate plans that the operator offers. Furthermore, usage habits are quite different for the various rate plans. However, a unique formula for each price plan is not feasible either, as some of these only contain a few customers. Therefore, we divided the entire customer base into 120 clusters, combining rate plans and bundles of similar characteristics (similar number of minutes, SMS and MB in the subscription), with a lower limit of 500 customers per cluster. Clustering algorithms were considered, however we had enough of clear structure in the data to avoid this.

The approach we use is similar to the network capacity model described in (Radosavljevik & van der Putten, 2014), where we discuss load pockets to simulate network service quality. Here, we treat each cluster as a revenue pocket. Therefore, we create a linear regression model for each cluster of customers taking various types of usage as input and revenue as output, and save it in a PMML format for scoring. This process is automated using KNIME. A screen shot of the process is shown in figure 1.

4. Deployment and results

We experimented with algorithms other than linear regression, such as support vector regression, random forests, regression trees, polynomial regression etc. However, these algorithms did not contribute significantly in reducing the total error and substantially increased the computation time.

A lesson learned here was to keep all tables in memory, instead of on disk. This reduced the run time of the total modeling process from one hour to 10 minutes. When using the more complex algorithms, the training for some clusters lasted longer than a few hours to calculate, especially when the sample size was large (more than 30,000 samples).

The goal of the model is to forecast service revenues for a full year. This is achieved using the following process. We start from the base of the current month and remove the customers marked for discontinuation. For each month, we first forecast the usage of the customer cluster. Then we add the projected inflow of customers, the changes in contract and mark customers who will discontinue their contract for each customer cluster. Last, but not least, we run this "new" base and usage figures through the regression models described in the previous section and predict the revenues of the next month. We repeat this 12 times.

From a run time perspective, our first run was too slow: 52 hours for 4 months prediction. We were able to reduce this to about 15 minutes per month by using R instead of KNIME for some data transformations, keeping everything in memory and only writing to disk once the full month is calculated, as well as optimizing the process flow of generating "new" random average customers for the inflow and contract changes and removing randomly selected customers from the base for the outflow. We display the forecasting results in Qlikview (2014) providing drill down opportunities per rate plan, channel, bundle etc. The business now has better insights into the forecast than the actuals, so there is request to update operational reporting on the actuals in the same way. To validate our approach we use typical prediction level measures such as RMSE or MAE. However, the key measure for the end users to accept the model was how close the sum of our predictions is to the actual revenues. On the total base our model was only 0.3% off the target, while the error of the standard budgeting process was 8 times higher.

In conclusion, we used mostly open source tools and simple algorithms in a complex deployment flow to optimize a key business problem. Our approach allows for testing of multiple scenarios for the inflow of customers, renewing contracts and outflow, as well as simulation of pricing the products differently. Its usability and rationale are currently being verified by the end users. Our approach can generalize to revenue simulation of any subscription based service with usage based pricing.

References

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K., & Wiswedel, B. (2007). KNIME: The Konstanz Information Miner. *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer.

Feenberg, D. R., Gentry, W. M., Gilroy, D., & Rosen, H. S. (1988). Testing the rationality of state revenue forecasts. National Bureau of Economic Research Cambridge, Mass., USA.

Fullerton, T. M. (1989). A composite approach to forecasting state government revenues: Case study of the idaho sales tax. *International Journal of Forecasting*, *5*, 373–380.

Harris, F. H. d. (2006). Keynote paper: Large-scale entry deterrence of a low-cost competitor: An early success of airline revenue management. *International Journal of Revenue Management*, *1*, 5–27.

QlikTech International AB (2014). Qlikview personal edition (version 11.2). `http://us-d.demo.qlik.com/download`.

R Development Core Team (2008). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Radosavljevik, D., & van der Putten, P. (2014). Large scale predictive modeling for micro-simulation of 3g air interface load. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1620–1629).

Teradata Corporation (2017). Teradata online library. `http://info.teradata.com/HTMLPubs/DB_TTU_15_00/index.html`.

Trueman, B., Wong, M. F., & Zhang, X.-J. (2001). Back to basics: Forecasting the revenues of internet firms. *Review of Accounting Studies*, *6(2–3)*, 305–329.

Predicting Termination of Housing Rental Agreements with Machine Learning

Michiel van Wezel M.WEZEL@DUDOKWONEN.NL
Dudok Wonen, Larenseweg 32, 1221 BW Hilversum, The Netherlands

Keywords: housing, real estate, asset management, prediction.

Abstract

Terminations of rental agreements (tenancy endings) are important in the business processes of housing associations. I describe the results of a model that predicts tenancy endings using Machine Learning.

1. Introduction

In the world of housing associations tenancy endings play an important role in all levels of business, e.g.:

- Rental Process: Ended tenancies lead to new rentals by new renters. This puts a burden on the staff responsible for renting out the houses.
- Maintenance: Dwellings must often be renovated before a new tenancy. This is costly, both in terms of effort and money.
- Sales: Often, part of a portfolio is labeled to be sold off to private owners upon tenancy ending. This takes effort from the responsible staff and brings in funds for new investments. Both must be planned in advance.
- Portfolio management: Knowing which dwellings become available and which renters have a (latent) desire to move helps in portfolio planning.
- Valuation: Dwellings are organized in clusters. The market value of each cluster is estimated yearly by a valuator. (This is a legal requirement for the financial statement.) To this end, the valuator uses a discounted cash flow (DCF) method. The rate of tenancy endings (abbreviated RTE) within a cluster plays an important role here: Higher rates of tenancy endings lead to increased values. This is caused by the decreased average discount period in case of a sell off.

In current practice, a moving average of the cluster-wise historical RTE is used to make prognoses. The parameter is of such importance that more accurate modeling is desirable.

2. Data & Model

Data from our Enterprise Resource Planning system and from the the yearly valuation process were collected over a three year period in a data vault. A data vault is basically a time-stamped representation of the underlying database of multiple information systems, enabling us to derive features from the evolution of these data. We extracted two data-sets from it. The first one contained data on (anonymised) renters, dwellings and contracts on reference date Jan 1-st 2014, augmented with an indicator for tenancy ending. The second one contained the same data on renters etc., but excludes the indicator. The data are highly noisy, with a lot of missings.

An ensemble classifier (see e.g., (Hastie et al., 2001)) consisting of 10000 decision trees was implemented in (R) (R Development Core Team, 2008).

3. Results

Based on a cross validation estimate, the model substantially outperforms the current practice of averaged historical RTE estimation. Figure 1 shows a clear lift for all sensitivity levels.

The trained model helps in understanding (or, at least hypothesizing) why tenancies end. Figure 2 shows the relative importances of the indicators used. Age of tenant and building type are the most important ones. The functional dependencies between the most impor-

Appearing in *Proceedings of Benelearn 2017*.

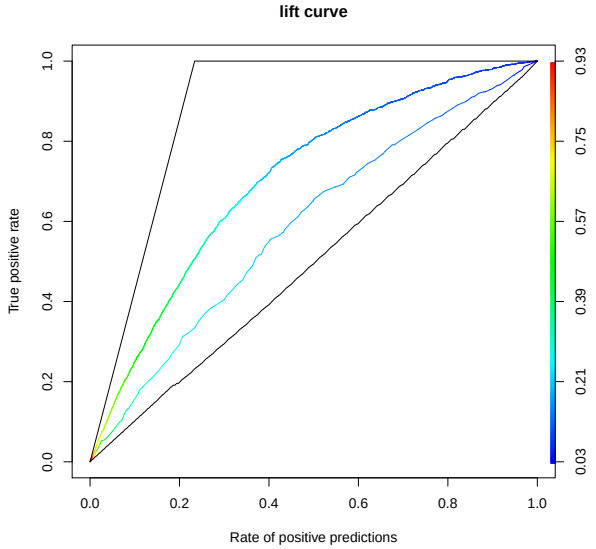


Figure 1. Lift curve depicting model performance of the ensemble model (top colored curve) versus the historical RTE estimate (bottom colored curve). The bottom black line depicts the baseline of a portfolio wide RTE.

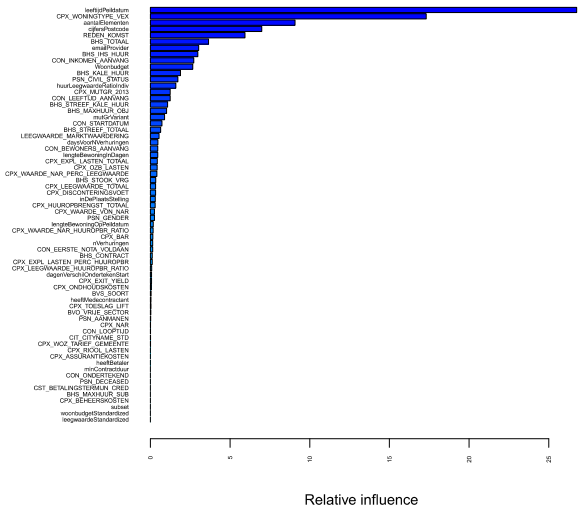


Figure 2. Relative importances of various indicators in the dataset.

tant indicators are visualized by partial dependence plots. (These average over the other indicators and must be interpreted with caution.) An example, showing the dependency of tenant age, is shown in Figure 3.

4. Conclusion

Application of machine learning techniques for predicting tenancy endings is viable. It allows for better planning than the currently used methods. It mitigates risks, lowers costs and potentially improves revenue. In the near future, we plan to improve our model by including more data and modeling a different outcome variable, i.e. period until tenancy end.

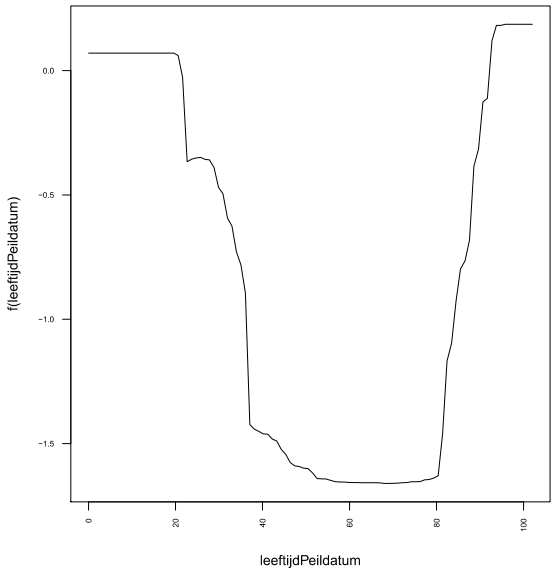


Figure 3. Partial dependency plot showing probability of tenancy ending versus tenant age.

References

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning.* Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

R Development Core Team (2008). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Anomaly Analytics and Structural Assessment in Process Industries

Martin Atzmueller M.ATZMULLER@UVT.NL
Tilburg University (TiCC), Warandelaan 2, 5037 AB Tilburg, The Netherlands

David Arnu DARNU@RAPIDMINER.COM
RapidMiner GmbH, Stockumer Straße 475, 44227 Dortmund, Germany

Andreas Schmidt SCHMIDT@CS.UNI-KASSEL.DE
University of Kassel (ITeG), Wilhelmshöher Allee 73, 34121 Kassel, Germany

Keywords: anomaly detection, exceptional model mining, sequential patterns, industry 4.0

Abstract

Detecting anomalous behavior can be of critical importance in an industrial application context: While modern production sites feature sophisticated alarm management systems, they mostly react to single events. In the context of process industries and heterogeneous data sources, we model sequential alarm data for anomaly detection and analysis, based on first-order Markov chain models. We outline hypothesis-driven and description-oriented modeling and provide an interactive dashboard for exploration and visualization.

1. Introduction

In many industrial areas, production facilities have reached a high level of automation: sensor readings are constantly analyzed and may trigger various forms of alarms. Then, the analysis of (exceptional) sequential patterns is an important task for obtaining insights into the process and for modelling predictive applications. The research project *Early detection and decision support for critical situations in production environments* (FEE) aims at detecting critical situations in production environments as early as possible and to support the facility operator in handling these situations, e. g., (Atzmueller et al., 2016a). Here, appropriate abstractions and analytics methods are necessary to adapt from a reactive to a proactive behavior.

Appearing in *Proceedings of Benelearn 2017*.

This paper summarizes the implementation of a comprehensive modeling and analytics approach for anomaly detection and analysis of heterogeneous data, as presented in (Atzmueller et al., 2017a).

2. Related Work

The investigation of sequential patterns and sequential trails are interesting and challenging tasks in data mining and network science, in particular in graph mining and social network analysis, e. g., (Atzmueller, 2014; Atzmueller, 2016b). In previous work (Atzmueller et al., 2016b), we have presented the DASHTrails approach that incorporates probability distributions for deriving transitions utilizing HypTrails (Singer et al., 2015). Based on that, the HypGraphs framework (Atzmueller et al., 2017b) provides a more general modeling approach. Using general weight-attributed network representations, we can infer transition matrices as graph interpretations.

Sequential pattern analysis has also been performed in the context of alarm management systems, where sequences are represented by the order of alarm notifications, e. g., (Folmer et al., 2014; Abele et al., 2013; Vogel-Heuser et al., 2015). In contrast to those approaches, we provide a systematic approach for the analysis of sequential transition matrices and its comparison relative to a set of hypotheses. Thus, similar to evidence networks in the context of social networks, e. g., (Mitzlaff et al., 2011) we model transitions assuming a certain interpretation of the data towards a sequential representation. Then, we can identify important influence factors.

3. Method

The detection and analysis of irregular or exceptional patterns, i.e., anomalies (Hawkins, 1980; Akoglu et al., 2015), in complex-structured heterogeneous data is a novel research area, e. g., for identifying new and/or emerging behavior, or for identifying detrimental or malicious activities. The former can be used for deriving new information and knowledge from the data, for identifying events in time or space, or for identifying interesting, important or exceptional groups.

In this paper, we focus on a combined detection and analysis approach utilizing heterogeneous data. That is, we include semi-structured, as well as structured data for enhancing the analysis. Furthermore, we also outline a description-oriented technique that does not only allow the detection of the anomalous patterns, but also its description using a given set of features. In particular, the concept of exceptional model mining, (Leman et al., 2008; Atzmueller, 2015; Duivesteijn et al., 2016) suitably enables such description-oriented approaches, adapting methods for the detection of interesting subgroups (that is, subgroup discovery) with more advanced target concepts for identifying exceptional (anomalous) groups. In our application context of an industrial production plants in an Industry 4.0 context, cf. (Vogel-Heuser et al., 2015; Folmer et al., 2017), we based our anomaly detection system on the analysis of the plant topology and alarm logs as well as on the similarity based analysis of metric sensor readings. The combined approach integrates both.

For sequential data, we formulate the "reference behavior" by collecting episodes of normal situations, which is typically observed for long running processes. Episodes of alarm sequences (formulated as hypotheses) can be compared to the normal situations in order to detect deviations, i.e., abnormal episodes. We map these sequences to transitions between functional units of an industrial plant. The results can also be used for diagnostics, by inspecting the transitions in detail. In summary, we utilize Bayesian inference on a first-order Markov chain model, see Figure 1. As an input, we provide a (data) matrix, containing the transitional information (frequencies) of transition between the respective states, according to the (observed) data. In addition, we utilize a set of hypotheses given by (row-normalized) stochastic matrices, modelling the given hypotheses. The estimation method outputs an evidence value, for each hypothesis, that can be used for ranking. Also, using the evidence values, we can compare the hypotheses in terms of their significance.

For modeling, we use the freely available RapidMiner (Mierswa et al., 2006) extension of HypGraphs,

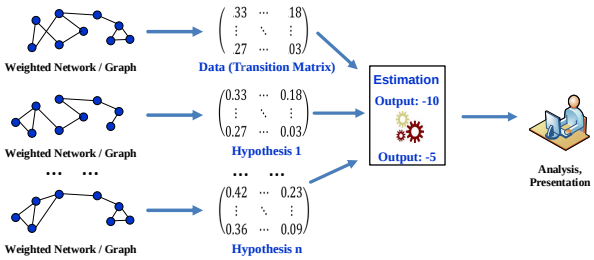


Figure 1. Overview on the modeling and analysis process.

that calculates the evidence values for different believe weights k and compares them directly with the given hypothesis and a random transition as a lower bound.

4. Process Model & Implementation

The first part of the analytical workflow is to build the transition network for training and testing the hypotheses. We build these hypotheses on real plant data and calculate the transition matrices for hourly time slots. In the same way, after further preprocessing (smoothing and down-sampling) we aggregate the corresponding raw sensor data. The calculated outlier score (Amer & Goldstein, 2012) is then presented, together with the evidence scores. A high outlier score indicates possible anomalous sensor readings and a low evidence score indicates deviating transition patterns in the alarm sequences. For further inspecting the outlier scores, we provide an additional dashboard. This shows the k highest outlier score for single sensor readings for a selected time segment and the associated sensor readings. Drilling-down from a high level of abstraction for a whole processing unit down to single sensor readings, a process engineer is then able to analyze possible critical situations in a convenient way.

For future work, we aim at extending the proposed approach by integrating the knowledge gained from a conceptual plant knowledge graph (Atzmueller et al., 2016a). We also plan to integrate the system into the Big data architecture proposed in (Klöpper et al., 2016), also considering further extensions on Big Data frameworks, e. g., (Meng et al., 2016; Carbone et al., 2015) and advanced assessment, exploration and explanation options, e. g., (Atzmueller et al., 2006; Atzmueller & Roth-Berghofer, 2010; Seipel et al., 2013) using advanced descriptive data analysis and modeling techniques, e. g., (Atzmueller, 2016a).

Acknowledgments

This work was partially funded by the BMBF project FEE under grant number 01IS14006.

References

Abele, L., Anic, M., Gutmann, T., Folmer, J., Kleinsteuber, M., & Vogel-Heuser, B. (2013). Combining Knowledge Modeling and Machine Learning for Alarm Root Cause Analysis. *MIM* (pp. 1843–1848).

Akoglu, L., Tong, H., & Koutra, D. (2015). Graph Based Anomaly Detection and Description. *DMKD*, *29*, 626–688.

Amer, M., & Goldstein, M. (2012). Nearest-Neighbor and Clustering-based Anomaly Detection Algorithms for Rapidminer. *Proc. RCOMM* (pp. 1–12).

Atzmueller, M. (2014). Analyzing and Grounding Social Interaction in Online and Offline Networks. *Proc. ECML-PKDD* (pp. 485–488). Springer.

Atzmueller, M. (2015). Subgroup Discovery. *WIREs: Data Mining and Knowledge Discovery*, *5*, 35–49.

Atzmueller, M. (2016a). Detecting Community Patterns Capturing Exceptional Link Trails. *Proc. IEEE/ACM ASONAM*. IEEE Press.

Atzmueller, M. (2016b). Local Exceptionality Detection on Social Interaction Networks. *Proc. ECML-PKDD* (pp. 485–488). Springer.

Atzmueller, M., Arnu, D., & Schmidt, A. (2017a). Anomaly Detection and Structural Analysis in Industrial Production Environments. *Proc. International Data Science Conference*. Salzburg, Austria.

Atzmueller, M., Baumeister, J., & Puppe, F. (2006). Introspective Subgroup Analysis for Interactive Knowledge Refinement. *Proc. AAAI FLAIRS* (pp. 402–407). Palo Alto, CA, USA: AAAI Press.

Atzmueller, M., Kloepper, B., Mawla, H. A., Jäschke, B., Hollender, M., Graube, M., Arnu, D., Schmidt, A., Heinze, S., Schorer, L., Kroll, A., Stumme, G., & Urbas, L. (2016a). Big Data Analytics for Proactive Industrial Decision Support. *atp edition*, *58*.

Atzmueller, M., & Roth-Berghofer, T. (2010). The Mining and Analysis Continuum of Explaining Uncovered. *Proc. AI-2010*. London, UK: SGAI.

Atzmueller, M., Schmidt, A., & Kibanov, M. (2016b). DASHTrails: An Approach for Modeling and Analysis of Distribution-Adapted Sequential Hypotheses and Trails. *Proc. WWW 2016 (Companion)*. ACM.

Atzmueller, M., Schmidt, A., Kloepper, B., & Arnu, D. (2017b). HypGraphs: An Approach for Analysis and Assessment of Graph-Based and Sequential Hypotheses. In *New Frontiers in Mining Complex Patterns*, LNAI. Springer.

Carbone, P., Ewen, S., Haridi, S., Katsifodimos, A., Markl, V., & Tzoumas, K. (2015). Apache Flink: Stream and Batch Processing in a Single Engine. *Data Engineering*, 28.

Duivesteijn, W., Feelders, A. J., & Knobbe, A. (2016). Exceptional Model Mining. *DMKD*, *30*, 47–98.

Folmer, J., Kirchen, I., Trunzer, E., Vogel-Heuser, B., Pötter, T., Graube, M., Heinze, S., Urbas, L., Atzmueller, M., & Arnu, D. (2017). Challenges for Big and Smart Data in Process Industries. *atp edition*, *01-02*.

Folmer, J., Schuricht, F., & Vogel-Heuser, B. (2014). Detection of Temporal Dependencies in Alarm Time Series of Industrial Plants. *Proc. 19th IFAC World Congress*, 24–29.

Hawkins, D. (1980). *Identification of Outliers*. London, UK: Chapman and Hall.

Klöpper, B., Dix, M., Schorer, L., Ampofo, A., Atzmueller, M., Arnu, D., & Klinkenberg, R. (2016). Defining Software Architectures for Big Data Enabled Operator Support Systems. *Proc. INDIN*.

Leman, D., Feelders, A., & Knobbe, A. (2008). Exceptional Model Mining. *Proc. ECML-PKDD* (pp. 1–16). Heidelberg, Germany: Springer.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., et al. (2016). MLLib: Machine Learning in Apache Spark. *JMLR*, *17*, 1–7.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). Yale: Rapid prototyping for complex data mining tasks. *Proc. KDD* (pp. 935–940). New York, NY, USA: ACM.

Mitzlaff, F., Atzmueller, M., Benz, D., Hotho, A., & Stumme, G. (2011). Community Assessment using Evidence Networks. *Analysis of Social Media and Ubiquitous Data*. Heidelberg, Germany: Springer.

Seipel, D., Köhler, S., Neubeck, P., & Atzmueller, M. (2013). Mining Complex Event Patterns in Computer Networks. In *New Frontiers in Mining Complex Patterns*, LNAI. Springer.

Singer, P., Helic, D., Hotho, A., & Strohmaier, M. (2015). Hyptrails: A Bayesian Approach for Comparing Hypotheses about Human Trails. *Proc. WWW*. New York, NY, USA: ACM.

Vogel-Heuser, B., Schütz, D., & Folmer, J. (2015). Criteria-based Alarm Flood Pattern Recognition Using Historical Data from Automated Production Systems (aPS). *Mechatronics*, *31*.