Experiment Design for Data Science – WS 2018/19
Exercise 2: Reproduce experimental results from a paper

Deadline for submission: Wed, January 16th, 2019, 23:55

For this assignment, you will work in **groups of three** and reproduce the experimental setup, experiments, and results as explained in a scientific paper. The task of interest is – based on careful experimental design – to confirm the numbers and findings reported in the paper or, alternatively, uncover inconsistencies and therefore challenge the conclusions drawn.

Specifically, by reproducing the experiments, it can be checked whether

- the information given in the paper is sufficient to reproduce the results reported
- statistically significant differences can be made out between the different settings, in particular when this is not reported in the paper (no significance tests, confidence intervals, p values, etc. or not even variance)
- the values reported in the paper could stem from the distribution sampled by the reproduced experiments

Your efforts need to be documented in a report that should be accompanied by the code and workflow you used to reproduce the experiments. To do so, first, identify the experimental setup and strategy taken in the paper and outline the steps necessary for you to reproduce results. Try to stay to the described steps as close as possible, i.e., use the same implementations and settings wherever possible. If the exact implementation is unknown or unavailable, find a reasonable substitute. You might have to consult additional sources to get the full picture of the used data or methods.

In each step, document which information is given in the paper and which measures you took to implement this step, i.e., which implementation and which parameters you used. Report the numbers obtained in each intermediate step. Identify deviations from the numbers reported in the original paper, their origin, and estimate whether they will have a significant impact on subsequent steps.

For the results, perform adequate tests to test for statistical significance. Justify your choice. Can you confirm the findings of the paper? Did you identify a flaw in their setup?

You need to produce the **results of one scientific paper out of a choice of three**. More details are found on the next page. The corresponding papers are attached to this document.

For writing the report, follow the ACM formatting guidelines, using the templates provided at https://www.acm.org/publications/proceedings-template. (Proceedings Style File: LaTeX2e - Strict Adherence to SIGS style; LaTeX recommended, but Word/OpenOffice is also ok). **Report page limit: Maximum 6 pages!** Focus on the key aspects!

In class on January 17th, 2019, each group will have to give a short **presentation of the work** (Strict time limit: 5 minutes!). To this end, prepare a deck of slides of 3-5 slides (including first slide containing group number, members, and chosen option),

presenting your strategies, encountered difficulties and key findings, as well as your conclusion about the experimental design, results, and conclusions presented in the paper.

Submit your report and presentation slides as PDFs together with any code, workflow, configurations, etc. you used, compressed in a ZIP archive via TUWEL.

Grading scheme:

- Inclusion of title, names, and registration numbers: max. 5%
- Quality of report, clarity of presentation: max. 10%
- Reproduction of results: max. 60%
- Description, interpretation, statistical testing: max. 25%

Option 1: Predicting the suitability of movies for an inflight viewing context

Based on the paper "TUD-MMC at MediaEval 2016: Context of Experience task" by Wang and Liem, your task is to reproduce their results on individual classifiers and combinations of classifiers (tables 1, 2, and 3).
The dataset referred to in the paper can be downloaded from https://www.dropbox.com/sh/j7nunecnzfjrp2r/AAC1BAf5JEv-rGUW9h02L2X2a?dl=0

Note that for this option, the main workload originates from feature selection, combination, and classifier configuration.

Option 2: Prediction of user demographics from music listening habits

Based on the paper "Prediction of User Demographics from Music Listening Habits" by Krismayer et al., your task is to reproduce their results on predicting different demographics using individual classifiers. Focus on the results shown in tables 1-4, i.e., not the experiments on subset size variation.

Note that for this option, the main workload originates from data filtering and data preparation.

Option 3: Classification of news and medical texts

Based on the paper "Text Categorization with Support Vector Machines: Learning with Many Relevant Features" by Joachims, your task is to reproduce the results on predicting categories of texts using different classifiers and in particular SVMs with different parameter settings. For the Reuters set, reproduce the results in Fig. 2, for Ohsumed the results mentioned in the text.

Note that for this option, the main workload originates from consulting additional sources, feature preparation, experiment setup, and classification optimization.

TUD-MMC at MediaEval 2016: Context of Experience task

Bo Wang
Delft University of Technology
Delft, The Netherlands
b.wang-6@student.tudelft.nl

Cynthia C. S. Liem
Delft University of Technology
Delft, The Netherlands
C.C.S.Liem@tudelft.nl

ABSTRACT

This paper provides a three-step framework to predict user assessment of the suitability of movies for an inflight viewing context. For this, we employed classifier stacking strategies. First of all, using the different modalities of training data, twenty-one classifiers were trained together with a feature selection algorithm. Final predictions were then obtained by applying three classifier stacking strategies. Our results reveal that different stacking strategies lead to different evaluation results. A considerable improvement can be found for the F1-score when using the label stacking strategy.

1. INTRODUCTION

A substantial amount of research has been conducted in recommender systems that focus on user preference prediction. Here, taking contextual information into account can have significant positive impact on the performance of recommender systems [1].

The MediaEval *Context of Experience* task focuses on a specific type of context: the viewing context of the user. The challenge considers predicting the multimedia content that users find most fitting to watch in a specific viewing condition, more specifically, while being on a plane.

2. DATASET DESCRIPTION AND INITIAL EXPERIMENTS

The dataset for the *Context of Experience* (CoE) task[5] contains metadata and pre-extracted features for 318 movies [6]. Features are multimodal and include textual features, visual features and audio features. The training set contains 95 labeled movies, which are labeled as 0 (bad for airplane) or 1 (good for airplane).

A set of initial experiments has been conducted in order to evaluate the usefulness of the various modalities in the *CoE* dataset [6]. A rule-based PART classifier was employed to evaluate the feature performance in terms of Precision, Recall and F1 Score, the result can be found in Table 1.

Features used	Precision	Recall	F1
User rating	0.371	0.609	0.461
Visual	0.447	0.476	0.458
Metadata	0.524	0.516	0.519
Metadata + user rating	0.581	0.6	0.583
Metadata + visual	0.584	0.6	0.586

Table 1: Results obtained by applying a rule-based PART classifier to the *Right Inflight* dataset.

3. MULTIMODAL CLASSIFIER STACKING

Ensemble learning uses a combination of different classifiers, usually getting a much better generalization ability. This particularly is the case for *weak learners*, which can be defined as learning algorithms that perform just slightly better than random guessing by themselves, but can be jointly grouped into an algorithm with arbitrarily high accuracy [2].

MediaEval 2016 Workshop, Oct. 20-21, 2016, Hilversum, Netherlands.

Therefore, we were interested in taking a multimodal classifier stacking approach to the given problem, and use a combination of multiple weak learners to 'boost' them into a strong learner.

The process can be separated into three stages: classifier selection, feature selection and classifier stacking.

3.1 Classifier Selection

First of all , we want to select base classifiers that will be useful candidates in a stacking approach. For this, we use the following classifier selection procedure:

1. Initialize a list of candidate classifiers. For each modality, we consider the following classifiers: k-nearest neighbor, nearest mean, decision tree, logistic regression, SVM, bagging, random forest, AdaBoost, gradient boosting, and naive Bayes. We do not apply parameter tuning, but take the default parameter values as offered by scikit-learn[1].

2. Perform 10-fold cross-validation on the classifiers. As input data, we use the training data set and its ground truth labels, per single modality. For the audio MFCC features, we set NaN values to 0, and calculate the average of each MFCC coefficient over all frames.

3. If Precision and Recall and F1-Score > 0.5, keep the candidate classifier on the given modality as base classifier for our stacking approach.

The selected base classifiers and their relevant modalities can be found in Table 2. It should be noted that the performance of `Bagging` and `Random forest` is not stable. This is because Bagging tries to use different subset of instances in each run and RandomForest tries to use different subsets of instances and features in each run.

3.2 Feature Selection

For each classifier and corresponding modality, a better-performing subspace of features may optimize results further. Since we have

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

Classifier	Modality	Precision	Recall	F1
k-Nearest neighbor	metadata	0.607	0.654	0.630
Nearest mean classifier	metadata	0.603	0.579	0.591
Decision tree	metadata	0.538	0.591	0.563
Logistic regression	metadata	0.548	0.609	0.578
SVM (Gaussian Kernel)	metadata	0.501	0.672	0.574
Bagging	metadata	0.604	0.662	0.631
Random Forest	metadata	0.559	0.593	0.576
AdaBoost	metadata	0.511	0.563	0.536
Gradient Boosting Tree	metadata	0.544	0.596	0.569
Naive Bayes	textual	0.545	0.987	0.702
k-Nearest neighbor	textual	0.549	0.844	0.666
SVM (Gaussian Kernel)	textual	0.547	1.000	0.707
k-Nearest neighbor	visual	0.582	0.636	0.608
Decision tree	visual	0.521	0.550	0.535
Logistic regression	visual	0.616	0.600	0.608
SVM (Gaussian Kernel)	visual	0.511	0.670	0.580
Random Forest	visual	0.614	0.664	0.638
AdaBoost	visual	0.601	0.717	0.654
Gradient Boosting Tree	visual	0.561	0.616	0.587
Logistic Regression	audio	0.507	0.597	0.546
Gradient Boosting Tree	audio	0.560	0.617	0.587

Table 2: Base classifier performance on multimodal dataset.

multiple learners, we employed the *Las Vegas Wrapper*(LVW) [3] feature selection algorithm for a feature subset selection. For each run, LVW generate a list of random features and evaluate the learner's error rate for n times, and select the best performing feature subspace as output.

In our case, we slightly modified LVW to optimize F1 score, where the original las vegas wrapper was developed for optimize accuracy.

For each base classifier, with the exception of the random forest classifier (as it already performs feature selection), we apply the LVW method, and achieve performance measures as listed in Table 2.

3.3 Classifier Stacking

In previous research, classifier stacking (or metalearning) has been proved beneficial for predictive performance by combining different learning systems which each have different inductive bias (e.g. representation, search heuristics, search space) [4]. By combining separately learned concepts, meta-learning is expected to derive a higher-level learned model that more accurately can predict than any of the individual learners. In our work, we consider three types of stacking strategies:

1. *Majority Voting:* this is the simplest case, where we select classifiers and feature subspaces through the steps above, and assign final predicted labels through majority voting on the labels of the 21 classifiers.

2. *Label Stacking:* Assume we have n instances and T base classifiers, then we can generate an n by T matrix consisting of predictions (labels) given by each classifier. Label combining strategy tries to build a second-level classifier based on this label matrix, and return a final prediction result for that.

3. *Label-Feature Stacking:* Similar to label stacking, label-feature stacking strategy uses both base-classifier predictions and features as training data to predict output.

4. RESULTS

We considered all prediction results by the 21 selected base classifiers, and then applied the three different classifier stacking strategies to the test data using 10-fold cross-validation. As results for label stacking vs. label attribute stacking were comparable on the training data, we only consider voting vs. label stacking on the test data.

All obtained results, on the training (development) and test dataset, are given in Table 3. On the training data, we notice significant improvement can be found in terms of Precision, Recall as well as F1 score in comparison to results obtained on individual modalities. The voting strategy results in the best precision score, but has bad performance in terms of recall. On the contrary, label stacking has higher recall and the highest F1 score.

Considering results obtained on the test dataset, we can conclude that *label stacking* is more robust than the voting strategy. For voting strategy, a significant decrease can be found in terms of precision on test set. This is because majority vote (and Bayesian averaging) tendency to over-fit derives from the likelihood's exponential sensitivity to random fluctuations in the sample, and increases with the number of models considered. Meanwhile, label stacking strategy performs reasonable well on test data.

Stacking Strategy	Precision	Recall	F1
Voting (cv)	0.94	0.57	0.71
Label Stacking (cv)	0.72	0.86	0.78
Label Attribute Stacking (cv)	0.71	0.79	0.75
Voting (test)	0.62	0.80	0.70
Label Stacking (test)	0.62	0.90	0.73

Table 3: Classifier Stacking results.

5. CONCLUSIONS

In our entry for the MediaEval CoE task, we aimed to improve classifier performance by a combination of classifier selection, feature selection and classifier stacking. Results reveal that employing a ensemble approach can considerably increase the classification performance, and is suitable for treating the multimodal *Right Inflight* dataset.

The larger diversity of base classifiers is able to produce a more robust ensemble classifier. On the other hand, a blending of multiple classifiers may also have some drawbacks, e.g computational costs, and difficulty in traceable interpretation.

We expect better results for our method can still be obtained through parameter tuning, and by applying more robust classifier stacking methods, such as feature weighted linear stacking [7].

6. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[2] Y. Freund and R. E. Schapire. A Decision-theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of computer and system sciences*, 55:119–139, 1997.

[3] H. Liu and R. Setiono. Feature selection and classification—a probabilistic wrapper approach. In *Proceedings of the 9th International Conference on Industrial and Engineering Applications of AI and ES*, pages 419–424, 1997.

[4] A. Prodromidis, P. Chan, and S. Stolfo. Meta-learning in distributed data mining systems: Issues and approaches. In

Advances in distributed and parallel knowledge discovery, pages 81–114. MIT/AAAI Press, 2000.

[5] M. Riegler, , C. Spampinato, M. Larson, P. Halvorsen, and C. Griwodz. The mediaeval 2016 context of experience task: Recommending videos suiting a watching situation. In *Proceedings of the MediaEval 2016 Workshop*, 2016.

[6] M. Riegler, M. Larson, C. Spampinato, P. Halvorsen, M. Lux, J. Markussen, K. Pogorelov, C. Griwodz, and H. Stensland. Right inflight? A dataset for exploring the automatic prediction of movies suitable for a watching situation. In *Proceedings of the 7th International Conference on Multimedia Systems*, pages 45:1–45:6. ACM, 2016.

[7] J. Sill, G. Takacs, L. Mackey, and D. Lin. Feature-weighted linear stacking. arXiv:0911.0460, 2009.

Prediction of User Demographics from Music Listening Habits

Thomas Krismayer
Christian Doppler Lab MEVSS
Institute for Software Systems Engineering
Johannes Kepler University Linz
thomas.krismayer@jku.at

Markus Schedl
Department of Computational Perception
Johannes Kepler University Linz
markus.schedl@jku.at

Peter Knees
Institute of Software Technology
and Interactive Systems
Vienna University of Technology
peter.knees@tuwien.ac.at

Rick Rabiser
Christian Doppler Lab MEVSS
Institute for Software Systems Engineering
Johannes Kepler University Linz
rick.rabiser@jku.at

ABSTRACT

Online activities such as social networking, shopping, and consuming multi-media create digital traces often used to improve user experience and increase revenue, e.g., through better-fitting recommendations and targeted marketing. We investigate to which extent the music listening habits of users of the social music platform Last.fm can be used to predict their age, gender, and nationality. We propose a TF-IDF-like feature modeling approach for artist listening information and artist tags combined with additionally extracted features. We show that we can substantially outperform a baseline majority voting approach and can compete with existing approaches. Further, regarding prediction accuracy vs. available listening data we show that even one single listening event per user is enough to outperform the baseline in all prediction tasks. We conclude that personal information can be derived from music listening information, which indeed can help better tailoring recommendations.

CCS CONCEPTS

•Computing methodologies → Machine learning approaches;
•Social and professional topics → *User characteristics;*

KEYWORDS

User Trait Prediction, Digital User Traces, Music Listening Habits

ACM Reference format:
Thomas Krismayer, Markus Schedl, Peter Knees, and Rick Rabiser. 2017. Prediction of User Demographics from Music Listening Habits. In *Proceedings of CBMI, Florence, Italy, June 19-21, 2017,* 7 pages.
DOI: 10.1145/3095713.3095722

CBMI, Florence, Italy

DOI: 10.1145/3095713.3095722

1 INTRODUCTION

Online activities such as using social networks or microblog services or shopping and consuming media leave digital traces, that indicate products or topics the user is interested in. These traces are recorded and many services use systems to recommend new items based on items the user selected or rated in the past (e.g., the music recommender system Last.fm or the online movie streaming service Netflix) [7].

It has been shown that many of the digital traces that are left by the users can also be exploited to predict additional information about them such as predicting a person's location from their tweets [1] or predicting personality traits from Facebook likes [10]. In this work, we focus on digital traces on the social music platform Last.fm, and use various different sources of information either directly from data available via the Last.fm API or extracted from the collected data to infer personal information of users.

We consider this a highly relevant topic with respect to digital media consumption and social media usage behavior for two reasons: on one hand, gaining a better understanding of the users will help in better understanding the contents of the media they are using, and thus help in creating more "semantic" indexing techniques. On the other hand, we are interested in how much this "harmless" and therefore often unthinkingly shared information can be used to derive additional information about the users. This second aspect exhibits direct ties to concerns regarding privacy and profiling.

To explore these aspects, we formulated the following two research questions: *(RQ1) To which extent is it possible to predict the age, gender, and nationality of the users based on their listening events and related information* (e. g., how the listening behavior changes over time)? *(RQ2) In which way does prediction accuracy depend on the available user data*, i. e., number of listening events?

The results of the proposed system can be utilized to enrich the input for recommender systems (e.g., to replace missing values for collaborative filtering approaches) or directly for recommending new items (e.g., artists that are popular in the country or within the age group of the user). In further steps the system could also be used to directly predict topics (e.g., genres) or items (e.g., artists or songs) the user is interested in, thus improving the user experience.

The remainder of this paper is structured as follows. In Section 2, we discuss literature related to the prediction of user traits from digital traces. Section 3 provides a description of the dataset used in

our experiments. We introduce the actual algorithm for predicting user traits in Section 4. In Section 5, we describe the experiments performed and the results gained. Finally, Section 6 wraps up the paper with a conclusion and an outlook on future work.

2 RELATED WORK

In this section, we discuss work on automated prediction of user traits from digital traces, structured according to the source of collected user traces.

Kosinski et al. [10] show that user traits can be predicted based on the **Facebook** Likes of a person. The predicted values include basic profile information, such as age and gender, but also highly personal attributes, such as sexual orientation, ethnicity, political views, and personality traits. The prediction is based on the Likes of 58,000 Facebook users, for which demographic profiles and psychometric tests are available. A follow-up study to [10], conducted by Youyou et al. [23], shows that personality judgments made from Facebook Likes can be even more accurate than those of close friends or family members. Golbeck et al. [2] show that the personality of Facebook users can even be predicted based only on their publicly available profile information.

The algorithm described by Cheng et al. [1] estimates the location of **Twitter** users based on the text of their tweets. The estimation is entirely content-based and does not rely on meta-data, such as profile or network information. The proposed algorithm is trained on Twitter users in continental USA whose locations are known and then predicts the user location by inferring probabilities for cities from the microblogs. In their experiment, Cheng et al. report that 51% of the users were placed within 100 miles of their actual hometown.

Most closely related to our paper is work that exploits **Last.fm** data to predict listener characteristics. Liu et al. [12] estimate the gender of Last.fm users based on their listening history. Additionally, the age is estimated in a binary form as under or above 24 years. The features for the classification are constructed purely from the listening events of the user and are based on three factors: the listening timestamps, the meta-data of the song and the artist (e.g., artist and song tags), as well as signal features of the songs. For both tasks, a support vector machine classifier (SVM) with RBF kernel is used and the average of five runs with 80% of the users as training set is reported. The accuracy for age is 71.1%; the accuracy for gender is 66.1%.

The approach described in the work by Wu et al. [22] estimates gender and age of Last.fm users based on music meta-data. Their algorithm uses the songs that the user most frequently listens to. In contrast to [12], the approach does not exploit temporal information, nor any audio-based features. The authors describe two different ways to generate features for the user: Term Frequency - Inverse Document Frequency (TF-IDF) combined with Latent Semantic Indexing (LSI) and Gaussian Super Vectors (GSV). For both tasks, SVM with RBF kernels are used in a two-fold cross validation. The reported accuracy for gender estimation is 78.87% and 78.21% for GSV and TF-IDF, respectively. For age estimation a mean absolute error of 3.69 and 4.25 is reported for the GSV and the TF-IDF approach, respectively.

In contrast to these two existing works [12, 22], our main *contributions* are: (i) we present a novel approach for the prediction of user traits from music listening habits that combines multiple sources of information and uses PCA-compressed TF-IDF-like features, (ii) we also support the prediction of user nationality, (iii) we ran our experiments with users with a very limited number of listening events, to assess performance in cold-start situations, and (iv) we compare different machine learning classification and regression algorithms.

3 DATASET

The dataset used in our experiments is a subset of the LFM-1b dataset [16]. It was created using the Last.fm API, which allows the collection of users' profile information (including age, gender, and country) as well as listening events for these users. Additionally we used weighted artists tags, which were also extracted using the Last.fm API and can be used to identify artists that produce similar music, for our experiments.

The LFM-1b dataset additionally includes scores describing the listening behavior of the users. These scores include novelty, i.e., percentage of new artists in a specific time period, mainstreaminess, i.e., how well the preferences of the user fit to the average preferences of all users, and different listening counts (e.g., the absolute number of distinct artists the user listened to, the average number of events per week, and the relative number of events for one specific day of the week).

Discarding from the LFM-1b dataset users with missing demographic information or less than 500 listening events, a total of 12,181 users remained for our experiments. This allows to use the same dataset for all three prediction tasks (age, gender, and country). The restriction to users with at least 500 listening events ensures that all users have the same number of listening events for the experiments with listening event subsets.

The dataset eventually contains users from 144 countries with 72.5% of them being male and the average age being 25.6 years. In terms of number of users, the top countries in our dataset are: USA (19% of all users), Russia (8.9%), Germany (8.4%), Brazil (7.9%), Poland (7.8%), Great Britain (7.8%), and the Netherlands (2.6%). This distribution is similar to the distribution among the users in the entire LFM-1b dataset.

3.1 Balanced Gender Dataset

Due to the high share of male users in the dataset the baseline for the accuracy of gender prediction is rather high (72.5%). Although the best classifiers perform significantly better (81.4%, cf. Section 5.4), it is difficult to assess the performance of these classifiers. To overcome this problem when investigating the first research question for gender, during all experiments for gender prediction, we created multiple datasets, for which the users are filtered by selecting all female users and randomly selecting exactly as many male users. The datasets resulting from this procedure contain a total of 6,698 users (compared to the 12,181 users of the entire dataset) with a 50% share of female users.

3.2 Sampling Listening Event Subsets

We sampled small random subsets from the listening histories of users with 1, 2, 5, 10, 20, 50, 100, 200, and 500 listening events per user to investigate our second research question, i. e., to what degree the accuracy of predictions depends on the number of listening events used for training.

4 PREDICTION OF USER TRAITS

For prediction of user traits, we developed three models, one for age, gender, and country, respectively. Each model is built individually and does not use results from the other models. Furthermore, the models are built entirely from the listening data of the users, metadata of the artists, and extracted user information. Therefore, e. g., for the prediction of age, the model does not use the gender or the country of the user.

4.1 Experimental Setup

The prediction models are evaluated with a 10-fold cross-validation on the dataset introduced in Section 3. All steps for the prediction pipeline (feature selection, feature vector generation, dimensionality reduction, classification/regression) were individually performed for the different user traits age, gender, and country. The calculations for all steps are based solely on the training set; this also implies that the selected features and the dimensionality reduction rules are different for each fold of the cross-validation.

4.2 Feature Selection

For each user, an individual feature vector is constructed containing elements from three separate sources – the first part is based on artist listening information, the second part on artist tag information, and the third part on additional user information provided as part of the LFM-1b dataset. These three parts are created independently from each other. The first two parts are vector normalized separately, for the third part this is pointless as we will explain below. Finally, the three parts are merged to create one feature vector per user ("early fusion").

The *first part* of a user's feature vector (*artist listening information*) is created as follows. 10,000 artists are selected based on the number of users that listened to them. The first half of artists that is selected are the artists that have the most different users in the overall training set that listened to them at least once. The second half of the artists is selected based on their number of different listeners in user-groups chosen for the specific task. This means the users in the training set are split into distinct groups and the artists with the most users listening to them for each of the groups are selected.

For the age prediction task the users are split into eight distinct age groups also used in [17]. These groups contain the users in the age intervals [6–17], [18–21], [22–25], [26–30], [31–40], [41–50], [51–60], and [61–100]. For the gender prediction the artists with the most male and female listeners, respectively, are selected. Finally for the country prediction task the groups comprise the countries with the most users in the training set. The dataset contains 144 different countries, however the feature selection only takes into account the 25 most common countries within the training data to concentrate on the most crucial user groups. For the whole dataset the 25 most common countries contain 88.5% of all users.

The *second part* of a user's feature vector (*artist tag information*) is created by selecting 10,000 tags in the same way as the artists for the first part of the vector. The tags with the most users that listened at least once to an artist associated with this tag (with a tag weight higher than 0) are selected. The first 5,000 tags are selected based on the overall training set, while the second half is selected based on the same user groups as for the artists.

The *third part* of the feature vector contains 42 *additional scores* for each user, comprising scores for novelty (i. e., how many new artists did the user listen to in a given time period), mainstreamness (i. e., how well do the genre preferences of the user fit to the overall genre preferences of all users in the dataset), and various listening event counts (e. g., the average number of listening events per week).

The differences in the range of the scores makes a vector normalization of the third part pointless. For instance, the novelty scores of a user are calculated in the interval [0–1], while the count values of listening events have no boundary and are often above 10,000.

4.3 Feature Vector Generation

The entries for the first two parts of the feature vector of a user are calculated in the form of TF-IDF values for a term t (i. e., an artist or a tag) and a document d (i. e., the listening history of this user) as:

$$\text{tf-idf}(t, d) = \left(1 + \log\left(f_{dt}\right)\right) \cdot \log\left(\frac{n}{f_t}\right) \tag{1}$$

where n is the number of users in the training set, and f_t is the number of users with at least one listening event with the artist or tag. While f_{dt} for artists simply is the number of listening events with the artists, the value for tags also takes the tag weight into account:

$$f_{dt} = \sum_{e \in E} weight(a_e, t') \tag{2}$$

where E is the listening history of the user, a_e is the artist of listening event e, and $weight(a_e, t')$ is the tag weight for tag t' and artist a_e, which is 0, if the artist is not connected to t'.

4.4 Dimensionality Reduction

The feature vectors that result from the previous step have a high dimensionality, therefore dimensionality reduction via Principal Component Analysis (PCA) [6] is performed. The PCA is performed on the combined first two parts of the feature vector (i. e., 20,000 dimensions) to ensure that correlations between artist and tag features can be resolved.

The number of features is thereby reduced from 20,000 to 450. The new number of features results from adding 50 features as long as the average variance gained per feature stays above 0.03% (i. e., 1.5% for the 50 new features). The dimensionality reduction is performed in Python using the library scikit-learn [13]. The transformation is calculated based solely on the training set. The compressed feature vectors for the test set are then constructed using the same transformation.

4.5 Predictions for Listening Event Subsets

For the predictions based on listening event subsets (cf. Section 3.2) only the PCA-compressed first and second part of the feature vectors is used. The third part of the vector includes information that is not available in a cold-start-like situation that is simulated with these experiments and can therefore not be used. For instance, the novelty score represents an indicator of how the listening behavior of the user changes over time – an information that the system cannot estimate for a user, who just has one single listening event.

The classification/regression algorithm is trained on the original user vectors containing all listening events for the users in the training set. Based on this model the predictions for all subsets of the test set are made.

5 EXPERIMENTS AND RESULTS

Based on the reduced feature vectors resulting from the dimensionality reduction, different supervised models are built. The models are constructed using a selection of diverse machine learning classifiers and regressors. For this purpose, we use the Java API of the open source library Weka [3]. In this section, we analyze the results for the individual experiments using the same evaluation methods as in [12, 22] (i. e., mean absolute error for age and accuracy for gender and nationality).

Additionally, we evaluate the performance of the best classifiers on the reduced listening event subsets and the datasets with balanced gender share. We compare the results for all tasks to a baseline to help interpret their quality.

5.1 Learning Algorithms

For the prediction of the results a variety of different supervised classification and regression techniques are used.

Support Vector Machines (SVMs) aim at separating two classes by defining a border function in a potentially higher dimensional space such that data points from the two classes lie on the different sides of the border. SVMs can also be used in regression tasks by creating a function such that all data points fall within a given maximum error margin. The values for new data points are then predicted with this function. The predictions are made using implementations of the Sequential Minimal Optimization algorithm (SMO [4, 9, 14] and SMOreg [18, 19]).

M5P [15, 21] is a decision tree algorithm enhanced with linear regression, which can be a decision criterion for some of the nodes within the tree. Based on this algorithm, **M5Rules** [5, 15, 21] creates a decision list that is filled with rules from decision trees built with M5P.

Linear regression generates a regression function as a linear combination of the features. Similarly **logistic regression** [11] predicts the class of a data point based on a linear combination of the features. We use the two logistic regression algorithms Bayesian Logistic Regression and Simple Logistic.

Naïve Bayes [8] and **DMNBtext** [20] use Bayes' theorem to predict the class of a new instance based on the probabilities for the different classes inferred from the training instances.

Table 1: Mean absolute error for age prediction (best results)

Classifier	Settings	Mean absolute error
SMOreg	RBF Kernel	4.13
SMOreg	Normalized Poly Kernel	4.17
SMOreg	Poly Kernel	4.20
Linear Regression		4.36
M5P		4.40
M5Rules		4.40
SMOreg	PUK	4.71
ZeroR		6.23

5.2 Baseline

The baseline for the given tasks represents a trivial lower bound for the results of the classifiers. For the classification tasks, the baseline used is a classifier that predicts the majority class of the training set for all instances of the test set. E. g., for country prediction the baseline is a classifier that predicts the country with the most users in the training set for all users in the test set. In case of a regression task, the classifier predicts the average value in the training set for all instances of the test set. For both cases the calculation is done with Weka's ZeroR classifier [3].

5.3 Age Prediction

Table 1 shows the algorithms that achieved the lowest mean absolute error for predicting the age of the users. The support vector regression (SMOreg) outperforms all other algorithms with three of the four kernels available for this task. The lowest error (4.1; achieved with the RBF kernel) is 66.3% of the error achieved with the baseline algorithm. The Linear Regression achieves a slightly better result than M5P and M5Rules. The baseline for this task is 6.2 (calculated with ZeroR).

The results for the age prediction based on the subsets of limited listening events can be seen in Figure 1. We achieved these results with the SMOreg algorithm using the RBF kernel, which produced the best results for the entire dataset. Just one single listening event is sufficient to predict the age of the user more accurately than the baseline approach (5.8 vs. 6.2). The error of the prediction decreases steadily with an increasing number of listening events. Also, the final prediction that uses all of the available listening events achieves an error even lower than the prediction based on 500 listening events per user.

5.4 Gender Prediction

The baseline for the gender prediction is 72.5%. As a result of each of the training folds having a majority of male users, this is the share of male users among the dataset (see Section 3). Table 2 shows the performance of the best classifiers for this task. The algorithm achieving the best results is the Bayesian Logistic Regression. This algorithm, which was developed for text categorization, benefits from the features of the feature vectors including clustered TF-IDF values, because TF-IDF weighting is an approach developed as basis for text analysis and text categorization. Both the support vector

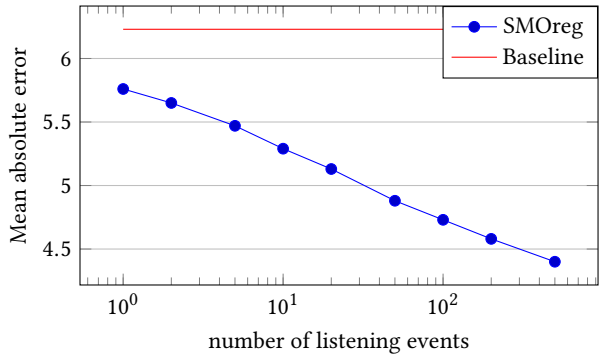

Figure 1: Error for age prediction on listening event subsets.

Table 2: Accuracy for gender prediction (best results)

Classifier	Settings	Accuracy
Bayesian Logistic Regression	Gaussian Prior	81.36%
SMO	Poly Kernel	81.24%
Simple Logistic		80.43%
SMO	Normalized Poly Kernel	78.06%
SMO	RBF Kernel	78.33%
DMNBtext		77.22%
SMO	PUK	76.31%
ZeroR		72.51%

classifier (SMO) and the logistic regression algorithm (Simple Logistic) achieve results very close to the Bayesian Logistic Regression. The other algorithms yield far lower accuracy.

Balanced gender dataset. To compensate for the uneven gender distribution in the dataset, datasets with uniform gender distributions have been created, as detailed in Section 3.1. In order to ensure that the experiments on this dataset are not influenced by the listeners that are randomly picked for classification, the filtering is performed five times, the experiments are performed on each of the resulting datasets, and results are reported averaged over the five runs.

Due to the resampling of the dataset to achieve equal distribution of gender, the baseline for this task is obviously 50%. The results for the three classifiers that performed best on the whole dataset are given in Table 3. We report the average and the standard deviation over the five runs. It can be seen that all three classifiers perform between 4.2% (SMO) and 4.5% (Simple Logistic) worse than the same classifiers trained on the whole dataset (cf. Table 2), but have to be compared to a much lower baseline. The accuracy for the SMO using a poly kernel is 154.0% relative to the new baseline; for the complete dataset the Bayesian Logistic Regression achieves a relative accuracy of only 112.2% compared to the baseline.

The average results for the five runs of the balanced gender subsets for the Bayesian Logistic Regression and the SMO can be seen in Figure 2. Both classifiers achieve very similar results for all listening event subsets and are able to achieve results better than the baseline with just one single listening event (up to 54.5% with

Table 3: Accuracy for gender prediction on the balanced dataset (best results)

Classifier	Settings	Accuracy
SMO	Poly Kernel	77.01% ± 0.30%
Bayesian Logistic Reg.	Gaussian Prior	76.91% ± 0.36%
Simple Logistic		75.88% ± 0.33%
Baseline		50%

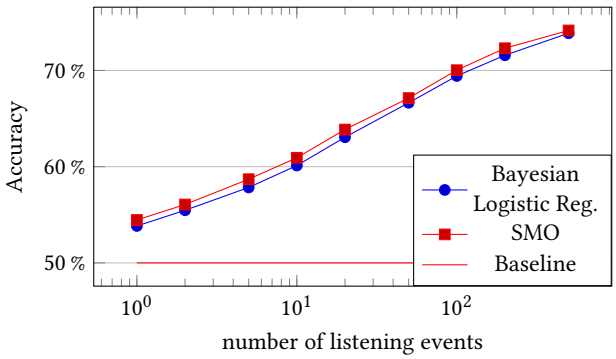

Figure 2: Accuracy for balanced gender prediction on listening event subsets.

Table 4: Accuracy for prediction of countries (best results)

Classifier	Settings	Accuracy
Simple Logistic		69.37%
SMO	Poly Kernel	69.36%
DMNBtext		63.11%
SMO	RBF Kernel	59.97%
SMO	Normalized Poly Kernel	59.57%
Naïve Bayes		57.39%
ZeroR		19.03%

the SMO classifier). The results improve steadily with additional listening events and also improve from 500 listening events to the overall result.

5.5 Country Prediction

Our third task is the prediction of the listeners' nationality. The baseline for this task is 19.0%, which equals the share of the most common country (USA) in the dataset. The classifiers that achieve the best results can be seen in Table 4. The two classifiers that perform best are the logistic regression algorithm (Simple Logistic) and the support vector classifier (SMO) which achieve 69.4% accuracy. The accuracy of the Simple Logistic algorithm is more than 3.6 times as high as the baseline.

The results for the reduced listening events can be seen in Figure 3, which include the results for the two best performing classifiers for the test set with all events (cf. Table 4). Similar to the predictions for age and for the balanced gender sets, both classifiers are able to beat the baseline with just one single listening

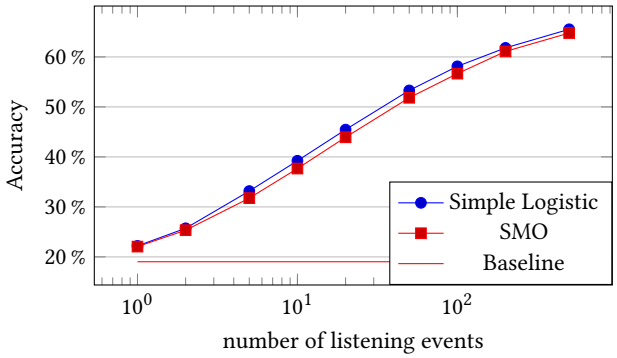

Figure 3: Accuracy for country prediction on listening event subsets.

event (22.2% accuracy for the SMO). and improve steadily with additional listening events.

5.6 Comparison with Existing Work

In the related work (cf. Section 2), the works of Liu et al. [12] and Wu et al. [22] have been introduced, which also target the prediction of user traits from music listening data.

The authors of [12] use the publicly available Last.fm 1K-users dataset to predict the gender and age of the users. This set contains users, for which user traits are missing. For the two experiments, the users, for which the respective trait is missing, are removed from the dataset. All the experiments are evaluated performing five runs with 80% of the users as training set and reporting the average of the results.

For this experiment we evaluated our approach with five-fold cross-validation, which also represents the average of five runs with 80% of the users as training set and additionally ensures that every user is part of the test set exactly once. Additional user information as in the Last.fm-1b dataset is not given and could therefore not be used.

For the gender prediction male users are removed from the dataset in order to create a set with a 50% share of female users. To lower the influence of the selected male users on the result we performed five runs of five-fold cross-validation – selecting different male users for each run – and reported the average result. The result achieved by our system is 72.9% (using Bayesian Logistic Regression with Gaussian prior), compared to an accuracy of 66.1%, which is the best result any approach in [12] achieved.

For the age prediction the authors split the user into the two classes “adolescents” (24 years and younger) and “adults” (25 years and older). Their best result achieved by [12] is 71.1%, compared to 72.4% achieved by our system (using Bayesian Logistic Regression with Laplace prior).

The authors of [22] use their own dataset to predict the age and gender of Last.fm users. Therefore it is unfortunately not possible to test our approach on their dataset; also the different number of users (96,807 vs. 12,181 users) and distribution of users (e. g., 66.2% vs. 72.5% male users) make a direct comparison of the received results pointless.

6 CONCLUSION AND OUTLOOK

Our experiments show that the listening history of a person allows to infer certain demographic information (RQ1). All three user traits age, gender, and country can be predicted to a substantial degree. For age the regression algorithm achieves an error that is 33.7% below the baseline error. For the balanced gender prediction and for the prediction of the nationality the increase in accuracy is 54.0% and 264.5% of the baseline, respectively. Even with a very small amount of listening events meaningful predictions can be made (RQ2). With increasing number of events the performance of the classifiers for all three user trait prediction tasks steadily increases.

Using the chosen approaches, we can indeed predict additional information about the users of online music listening services, solely from their listening histories. While the broad categorizations can help in tailoring collaborative as well as content-based recommender systems to their user groups, given the shown current limitations, however, it seems unlikely to generally predict personal information about the users that can affect their privacy.

As part of future work, we will consider additional listener- and listening-related aspects, for instance, exploiting the temporal information attached to listening events in greater depth. Also content-based features could be extracted and investigated, provided the respective audio is available. Another area that could be targeted in future work is deep learning – in addition to the learning algorithms used in our evaluation presented in this work.

ACKNOWLEDGMENTS

This work has been supported by the Christian Doppler Forschungsgesellschaft, Austria and Primetals Technologies.

REFERENCES

[1] Z. Cheng, J. Caverlee, and K. Lee. 2010. You Are Where You Tweet: A Content-based Approach to Geo-locating Twitter Users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 759–768.

[2] J. Golbeck, C. Robles, and K. Turner. 2011. Predicting Personality with Social Media. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*. ACM, 253–262.

[3] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations* 11, 1 (2009), 10–18.

[4] T. Hastie and R. Tibshirani. 1998. Classification by Pairwise Coupling. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*. MIT Press.

[5] G. Holmes, M. Hall, and E. Frank. 1999. Generating Rule Sets from Model Trees. In *Proceedings of the 12th Australian Joint Conference on Artificial Intelligence*. Springer, 1–12.

[6] H. Hotelling. 1933. Analysis of a Complex of Statistical Variables Into Principal Components. *Journal of Educational Psychology* 24, 6 (1933), 417–441 and 498–520.

[7] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 8th IEEE International Conference on Data Mining*. IEEE, 263–272.

[8] G. John and P. Langley. 1995. Estimating Continuous Distributions in Bayesian Classifiers. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 338–345.

[9] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. 2001. Improvements to Platt’s SMO Algorithm for SVM Classifier Design. *Neural Computation* 13, 3 (2001), 637–649.

[10] M. Kosinski, D. Stillwell, and T. Graepel. 2013. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences* 110, 15 (2013), 5802–5805.

[11] S. le Cessie and J.C. van Houwelingen. 1992. Ridge Estimators in Logistic Regression. *Applied Statistics* 41, 1 (1992), 191–201.

[12] J. Liu and Y. Yang. 2012. Inferring Personal Traits from Music Listening History. In *Proceedings of the 2nd International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies*. ACM, 31–36.

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

[14] J. Platt. 1998. Fast Training of Support Vector Machines using Sequential Minimal Optimization. In *Advances in Kernel Methods - Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smola (Eds.). MIT Press.

[15] R. Quinlan. 1992. Learning with Continuous Classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*. World Scientific, 343–348.

[16] M. Schedl. 2016. The LFM-1b Dataset for Music Retrieval and Recommendation. In *Proceedings of the ACM International Conference on Multimedia Retrieval*. ACM, 103–110.

[17] M. Schedl, D. Hauger, K. Farrahi, and M. Tkalčič. 2015. On the Influence of User Characteristics on Music Recommendation. In *Proceedings of the 37th European Conference on Information Retrieval*. Springer.

[18] S.K. Shevade, S.S. Keerthi, C. Bhattacharyya, and K.R.K. Murthy. 1999. Improvements to the SMO Algorithm for SVM Regression. *IEEE Transactions on Neural Networks* 11 (1999), 1188–1193.

[19] A.J. Smola and B. Schoelkopf. 1998. *A tutorial on support vector regression*. Technical Report. NeuroCOLT2 Tech. Rep. NC2-TR-1998-030.

[20] J. Su, H. Zhang, C. Ling, and S. Matwin. 2008. Discriminative Parameter Learning for Bayesian Networks. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, 1016–1023.

[21] Y. Wang and I.H. Witten. 1997. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*. Springer.

[22] M. Wu, J. Jang, and C. Lu. 2014. Gender Identification and Age Estimation of Users Based on Music Metadata. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*. ISMIR, 555–560.

[23] W. Youyou, M. Kosinski, and D. Stillwell. 2015. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences* 112, 4 (2015), 1036–1040.

Text Categorization with Support Vector Machines: Learning with Many Relevant Features

Thorsten Joachims

Universität Dortmund
Informatik LS8, Baroper Str. 301
44221 Dortmund, Germany

Abstract. This paper explores the use of Support Vector Machines (SVMs) for learning text classifiers from examples. It analyzes the particular properties of learning with text data and identifies why SVMs are appropriate for this task. Empirical results support the theoretical findings. SVMs achieve substantial improvements over the currently best performing methods and behave robustly over a variety of different learning tasks. Furthermore, they are fully automatic, eliminating the need for manual parameter tuning.

1 Introduction

With the rapid growth of online information, text categorization has become one of the key techniques for handling and organizing text data. Text categorization techniques are used to classify news stories, to find interesting information on the WWW, and to guide a user's search through hypertext. Since building text classifiers by hand is difficult and time-consuming, it is advantageous to learn classifiers from examples.

In this paper I will explore and identify the benefits of *Support Vector Machines (SVMs)* for text categorization. SVMs are a new learning method introduced by V. Vapnik et al. [9] [1]. They are well-founded in terms of computational learning theory and very open to theoretical understanding and analysis.

After reviewing the standard feature vector representation of text, I will identify the particular properties of text in this representation in section 4. I will argue that SVMs are very well suited for learning in this setting. The empirical results in section 5 will support this claim. Compared to state-of-the-art methods, SVMs show substantial performance gains. Moreover, in contrast to conventional text classification methods SVMs will prove to be very robust, eliminating the need for expensive parameter tuning.

2 Text Categorization

The goal of text categorization is the classification of documents into a fixed number of predefined categories. Each document can be in multiple, exactly one, or no category at all. Using machine learning, the objective is to learn classifiers

from examples which perform the category assignments automatically. This is a supervised learning problem. Since categories may overlap, each category is treated as a separate binary classification problem.

The first step in text categorization is to transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm and the classification task. Information Retrieval research suggests that word stems work well as representation units and that their ordering in a document is of minor importance for many tasks. This leads to an attribute-value representation of text. Each distinct word[1] w_i corresponds to a feature, with the number of times word w_i occurs in the document as its value. To avoid unnecessarily large feature vectors, words are considered as features only if they occur in the training data at least 3 times and if they are not "stop-words" (like "and", "or", etc.).

This representation scheme leads to very high-dimensional feature spaces containing 10000 dimensions and more. Many have noted the need for feature selection to make the use of conventional learning methods possible, to improve generalization accuracy, and to avoid "overfitting". Following the recommendation of [11], the *information gain* criterion will be used in this paper to select a subset of features.

Finally, from IR it is known that scaling the dimensions of the feature vector with their *inverse document frequency (IDF)* [8] improves performance. Here the "tfc" variant is used. To abstract from different document lengths, each document feature vector is normalized to unit length.

3 Support Vector Machines

Support vector machines are based on the *Structural Risk Minimization* principle [9] from computational learning theory. The idea of structural risk minimization is to find a hypothesis h for which we can guarantee the lowest true error. The true error of h is the probability that h will make an error on an unseen and randomly selected test example. An upper bound can be used to connect the true error of a hypothesis h with the error of h on the training set and the complexity of H (measured by VC-Dimension), the hypothesis space containing h [9]. Support vector machines find the hypothesis h which (approximately) minimizes this bound on the true error by effectively and efficiently controlling the VC-Dimension of H.

SVMs are very **universal learners.** In their basic form, SVMs learn linear threshold function. Nevertheless, by a simple "plug-in" of an appropriate kernel function, they can be used to learn polynomial classifiers, radial basic function (RBF) networks, and three-layer sigmoid neural nets.

One remarkable property of SVMs is that their ability to learn can be **independent of the dimensionality of the feature space.** SVMs measure the complexity of hypotheses based on the margin with which they separate the

[1] The terms "word" and "word stem" will be used synonymously in the following.

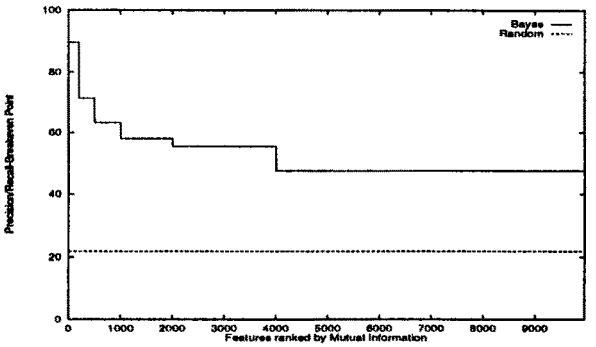

Fig. 1. Learning without using the "best" features.

data, not the number of features. This means that we can generalize even in the presence of very many features, if our data is separable with a wide margin using functions from the hypothesis space.

The same margin argument also suggest a heuristic for **selecting good parameter settings** for the learner (like the kernel width in an RBF network) [9]. The best parameter setting is the one which produces the hypothesis with the lowest VC-Dimension. This allows fully automatic parameter tuning without expensive cross-validation.

4 Why Should SVMs Work Well for Text Categorization?

To find out what methods are promising for learning text classifiers, we should find out more about the properties of text.

High dimensional input space: When learning text classifiers, one has to deal with very many (more than 10000) features. Since SVMs use overfitting protection, which does not necessarily depend on the number of features, they have the potential to handle these large feature spaces.

Few irrelevant features: One way to avoid these high dimensional input spaces is to assume that most of the features are irrelevant. Feature selection tries to determine these irrelevant features. Unfortunately, in text categorization there are only very few irrelevant features. Figure 1 shows the results of an experiment on the Reuters "acq" category (see section 5). All features are ranked according to their (binary) information gain. Then a naive Bayes classifier [2] is trained using only those features ranked 1-200, 201-500, 501-1000, 1001-2000, 2001-4000, 4001-9962. The results in figure 1 show that even features ranked lowest still contain considerable information and are somewhat relevant. A classifier using only those "worst" features has a performance much better than random. Since it seems unlikely that all those features are completely redundant, this leads to the conjecture that a good classifier should combine many features (learn a "dense" concept) and that aggressive feature selection may result in a loss of information.

Document vectors are sparse: For each document, the corresponding document vector contains only few entries which are not zero. Kivinen et al. [4] give both theoretical and empirical evidence for the mistake bound model that "additive" algorithms, which have a similar inductive bias like SVMs, are well suited for problems with dense concepts and sparse instances.

Most text categorization problems are linearly separable: All Ohsumed categories are linearly separable and so are many of the Reuters (see section 5) tasks. The idea of SVMs is to find such linear (or polynomial, RBF, etc.) separators.

These arguments give theoretical evidence that SVMs should perform well for text categorization.

5 Experiments

The following experiments compare the performance of SVMs using polynomial and RBF kernels with four conventional learning methods commonly used for text categorization. Each method represents a different machine learning approach: density estimation using a naive Bayes classifier [2], the Rocchio algorithm [7] as the most popular learning method from information retrieval, a distance weighted k-nearest neighbor classifier [5][10], and the C4.5 decision tree/rule learner [6]. SVM training is carried out with the SVM^{light} [2] package. The SVM^{light} package will be described in a forthcoming paper.

Test Collections: The empirical evaluation is done on two test collection. The first one is the "ModApte" split of the Reuters-21578 dataset compiled by David Lewis. The "ModApte" split leads to a corpus of 9603 training documents and 3299 test documents. Of the 135 potential topic categories only those 90 are used for which there is at least one training and one test example. After preprocessing, the training corpus contains 9962 distinct terms.

The second test collection is taken from the Ohsumed corpus compiled by William Hersh. From the 50216 documents in 1991 which have abstracts, the first 10000 are used for training and the second 10000 are used for testing. The classification task considered here is to assign the documents to one or multiple categories of the 23 MeSH "diseases" categories. A document belongs to a category if it is indexed with at least one indexing term from that category. After preprocessing, the training corpus contains 15561 distinct terms.

Results: Figure 2 shows the results on the Reuters corpus. The *Precision/Recall-Breakeven Point* (see e. g. [3]) is used as a measure of performance and *microaveraging* [10][3] is applied to get a single performance value over all binary classification tasks. To make sure that the results for the conventional methods are not biased by an inappropriate choice of parameters, all four methods were run after selecting the 500 best, 1000 best, 2000 best, 5000 best, (10000 best,) or all features using information gain. At each number of features the values $\beta \in \{0, 0.1, 0.25, 0.5, 1.0\}$ for the Rocchio algorithm and $k \in \{1, 15, 30, 45, 60\}$

[2] http://www-ai.informatik.uni-dortmund.de/thorsten/svm_light.html

					SVM (poly) degree d =					SVM (rbf) width γ =			
	Bayes	Rocchio	C4.5	k-NN	1	2	3	4	5	0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined: **86.0**					combined: **86.4**			

Fig. 2. Precision/recall-breakeven point on the ten most frequent Reuters categories and microaveraged performance over all Reuters categories. k-NN, Rocchio, and C4.5 achieve highest performance at 1000 features (with $k = 30$ for k-NN and $\beta = 1.0$ for Rocchio). Naive Bayes performs best using all features.

for the k-NN classifier were tried. The results for the parameters with the best performance on the test set are reported.

On the Reuters data the k-NN classifier performs best among the conventional methods (see figure 2). This replicates the findings of [10]. Compared to the conventional methods all SVMs perform better independent of the choice of parameters. Even for complex hypotheses spaces, like polynomials of degree 5, no overfitting occurs despite using all 9962 features. The numbers printed in bold in figure 2 mark the parameter setting with the lowest VCdim estimate as described in section 3. The results show that this strategy is well-suited to pick a good parameter setting automatically and achieves a microaverage of 86.0 for the polynomial SVM and 86.4 for the RBF SVM. With this parameter selection strategy, the RBF support vector machine is better than k-NN on 63 of the 90 categories (19 ties), which is a significant improvement according to the binomial sign test.

The results for the Ohsumed collection are similar. Again k-NN is the best conventional method with a microaveraged precision/recall-breakeven point of 59.1. C4.5 fails on this task (50.0) and heavy overfitting is observed when using more than 500 features. Naive Bayes achieves a performance of 57.0 and Rocchio reaches 56.6. Again, with 65.9 (polynomial SVM) and 66.0 (RBF SVM) the SVMs perform substantially better than all conventional methods. The RBF SVM outperforms k-NN on all 23 categories, which is again a significant improvement.

Comparing training time, SVMs are roughly comparable to C4.5, but they are more expensive than naive Bayes, Rocchio, and k-NN. Nevertheless, current research is likely to improve efficiency of SVM-type quadratic programming

problems. SVMs are faster than k-NN at classification time. More details can found in [3].

6 Conclusions

This paper introduces support vector machines for text categorization. It provides both theoretical and empirical evidence that SVMs are very well suited for text categorization. The theoretical analysis concludes that SVMs acknowledge the particular properties of text: (a) high dimensional feature spaces, (b) few irrelevant features (dense concept vector), and (c) sparse instance vectors.

The experimental results show that SVMs consistently achieve good performance on text categorization tasks, outperforming existing methods substantially and significantly. With their ability to generalize well in high dimensional feature spaces, SVMs eliminate the need for feature selection, making the application of text categorization considerably easier. Another advantage of SVMs over the conventional methods is their robustness. SVMs show good performance in all experiments, avoiding catastrophic failure, as observed with the conventional methods on some tasks. Furthermore, SVMs do not require any parameter tuning, since they can find good parameter settings automatically. All this makes SVMs a very promising and easy-to-use method for learning text classifiers from examples.

References

1. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, November 1995.
2. T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *International Conference on Machine Learning (ICML)*, 1997.
3. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. Technical Report 23, Universität Dortmund, LS VIII, 1997.
4. J. Kivinen, M. Warmuth, and P. Auer. The perceptron algorithm vs. winnow: Linear vs. logarithmic mistake bounds when few input variables are relevant. In *Conference on Computational Learning Theory*, 1995.
5. T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
6. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
7. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall Inc., 1971.
8. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
9. Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.
10. Y. Yang. An evaluation of statistical approaches to text categorization. Technical Report CMU-CS-97-127, Carnegie Mellon University, April 1997.
11. Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *International Conference on Machine Learning (ICML)*, 1997.