A benchmarking study of classification techniques for behavioral data

Journal of Data Science and Analytics manuscript No.
(will be inserted by the editor)

A Benchmarking Study of Classification Techniques for Behavioral Data

Sofie De Cnudde · David Martens · Theodoros Evgeniou · Foster Provost

the date of receipt and acceptance should be inserted later

Abstract The predictive power of increasingly common large-scale, behavioral data has been emphasized by previous academic research. Such data captures human behavior through the actions and/or interactions of people. Its sparsity and ultra-high dimensionality pose significant challenges to state-of-the-art classification techniques. Moreover, no consensus exists regarding the choice of methods that make a feasible trade-off between classification performance and computational expense. This paper provides a contribution in this direction through a systematic benchmarking study. Forty-one fine-grained behavioral data sets are analyzed with 11 classifiers. Statistical performance comparisons enriched with learning curve analyses demonstrate two important findings. First, an inherent generalization performance vs. time trade-off becomes clear, making the choice for an appropriate classifier dependent on computation constraints and data set characteristics. Logistic regression achieves the best AUC, however it takes the longest time to train. An associated result is that L2 regularization proves better than sparse L1 regularization. An attractive generalization/time trade-off is achieved by a similarity-based technique (PSN). Second, although the data sets used are large, the results illustrate that as a direct consequence of its high-dimensionality and sparseness, significant value lies in collecting and analyzing even more data. This finding is observed both in the instance and in the feature dimensions, contrasting with learning curve studies on traditional data. The results of this study provide guidance for researchers and practitioners for the selection of appropriate classification techniques, sample sizes and data features, while also providing focus in scalable algorithm design in the face of large, behavioral data.

Keywords comparative study · classification · big behavioral data · high-dimensional · sparse

S. De Cnudde
Department of Engineering Management
University of Antwerp
Antwerp, Belgium
E-mail: sofie.decnudde@uantwerpen.be

D. Martens
Department of Engineering Management
University of Antwerp
Antwerp, Belgium
E-mail: david.martens@uantwerpen.be

T. Evgeniou
INSEAD
Fontainebleau, France
E-mail: theodoros.evgeniou@insead.edu

F. Provost
Department Information, Operations & Management Sciences
Stern School of Business
New York University
New York, USA
E-mail: fprovost@stern.nyu.edu

1 Introduction

This paper focuses on very large-scale behavioral data (Chen et al., 2009), which have become increasingly common as the subject of analysis over the past two decades, as more of people's activities are recorded and quantified. Following the definition by Shmueli (2016), big

behavioral data captures human behavior through the actions and/or interactions of people. These form a record of a person's behavior captured as fine-grained features. Customer transactions with a bank, web surfers' web visiting behavior, mobile phone users' visited locations, and Facebook Likes are just a few examples. Predictive modeling based on behavioral data has demonstrated promising result. Such data can be telling of a person's personality traits (Kosinski et al., 2013), his interest in banking products (Martens et al., 2016), his interest in a news article (Liu et al., 2010), his interest in a (mobile) ad (Li and Du 2012; Perlich et al. 2014), his tendency to churn (Verbeke et al., 2014), his credit default behavior (De Cnudde et al., 2015) or his tendency to commit fraudulent activities (Fawcett and Provost 1997; Junqué de Fortuny et al. 2014).

In fine-grained behavior data, behavior is represented via the presence or absence of an action-having-been-taken (binary), or in more detail by the strength or the frequency of each individual action (numeric). An important characteristic of fine-grained behavior data is that the set of all possible behaviors (features) an entity can exhibit is enormous (such as the set of all possible locations or webpages one can visit), resulting in ultra-high dimensional data. Moreover, there is a limit on a person's so-called behavioral capital (Junqué de Fortuny et al., 2013), how many behaviors they can reasonably engage in; the result is that among all possible actions represented by fine-grained features, a person will exhibit relatively few. This results in extremely sparse data. The high-dimensionality and sparsity stand in stark contrast to data represented by traditional sociodemographic features or summarizing features such as RFM (recency, frequency, monetary) values.

In spite of the growing availability of big behavioral data (Yang and Wu, 2006), its potential for social science research (Shmueli, 2016), and the numerous studies clearly demonstrating its value for predictive purposes, such data poses significant challenges for traditional state-of-the-art data mining techniques (Provost and Kolluri 1999; Brain and Webb 2002; Dalessandro 2013). One such challenge is the curse of dimensionality (Donoho, 2000): large numbers of features results in a highly sparse and highly scattered data space, making it very difficult to calculate similarity or to capture general patterns. Researchers have coped with such challenges by either scaling up the classifiers (for example, Tsang et al. 2005; Collobert et al. 2006; Nie et al. 2014) or scaling down the data dimensionality (for example, Chang et al. 2010; Tan et al. 2014). The latter can be done through summarizing the fine-grained features in a manner similar to RFM (De Bock and Van den Poel, 2010) or through dimensionality reduction techniques (Kosinski et al., 2013). Using behavioral summaries has shown to result in lower predictive performance in comparison to using the features with their full granularity (De Cnudde et al. 2015; Martens et al. 2016). Matrix factorization-based dimensionality reduction and hashing techniques (Weinberger et al. 2009;Li et al. 2012) can be computationally efficient when faced with large, high-dimensional data with respect to time and space usage. However, Clark and Provost (2016) demonstrate that care should be taken in employing them with behavioral data since using the full feature set effectively results in higher predictive performance compared to a reduced feature set. In other high-dimensional contexts such as text classification, using all fine-grained features has also resulted in the best generalization performance (Joachims 1998; Li et al. 2012). The question that this paper addresses is whether and to what extend widely used, traditional classifiers can cope with this complex and rich type of data (Brain and Webb 2002; Wu et al. 2014).

Currently, no clear consensus has been reached in the literature regarding which classifier to employ for such data (see Table 1). Most of these studies start with a data-centric perspective, examining one or two data sets using one or more classification techniques. This is done either to demonstrate the predictive power present in a data set, to compare existing techniques, or to benchmark a self-developed technique against state-of-the-art classifiers. However, most papers do not provide clear-cut explanations as to why a certain technique is elected over others for analysis. Thus, more specifically, the present paper (1) helps provide guidance on the selection of an appropriate classification method, and (2) provides an assessment of the techniques' robustness.

Regarding guidance (1), benchmarking studies such as this are useful for comparing the

performance of a collection of techniques – comparing them in a systematic manner promotes statistically sound conclusions on the one hand and on the other hand leads to practical guidelines directing researchers and practitioners to an appropriate technique suited to their needs. In the past, large-scale benchmarking studies of data mining algorithms have been performed (for example, King et al. 1995; Lim et al. 2000; Meyer et al. 2002; Michie et al. 2009; Fernández-Delgado et al. 2014). Benchmarking is also often carried out between two or more techniques, investigating when which technique performs better (for example, Langley et al. 1992;Ralaivola and d'Alché Buc 2001;Huang et al. 2003;Perlich et al. 2003). However, to our knowledge, no comprehensive comparative study has yet been done focusing specifically on massive, sparse behavioral data, even though they are becoming common in applications of machine learning. This benchmarking study follows in the tradition of Forman (2003) and Fernández-Delgado et al. (2014), and we follow the advice of Demšar (2006), among others.

Regarding robustness (2), we also study the performance of the classification algorithms under varying training set sizes. This is done with learning curves, which investigate the impact of data size on classification performance (Perlich et al., 2003). From this analysis, conclusions can be drawn regarding the extent to which the techniques scale up in terms of performance for increasing data set size, in both the instance and feature dimensions. It is important to understand if and when more data leads to better predictive performance: organizations must plan their investment in collecting and storing even more data, and practitioners should have an idea about how the results of a pilot study on a data subset are likely to translate into results on a later, much larger production data set. A starting point for learning curve analysis in behavioral data was given in Junqué de Fortuny et al. (2013), which we will expand in a systematic manner. Importantly, the learning curves for behavioral data show strikingly different behavior from learning curve studies on more traditional data (Perlich et al., 2003).

In summary, the contributions of this paper are as follows:

I We perform a comparative analysis of state-of-the-art classification techniques on behavioral data sets. We compare both the predictive and computational performance for significant differences. Subsequently, recommendations are formulated to guide the choice of a predictive technique when confronted with behavioral data.

II We also assess the predictive value of behavioral data depending on two different data modeling schemes. An analysis is performed regarding the (un)importance of the strength of a behavioral action (binary vs. numeric data) for the analyzed techniques. Hence, guidance is offered regarding how to model behavioral data so as to reach optimal performance.

III The third contribution is a learning curve analysis of the classification techniques such that performance patterns become clear under changing data set sizes. The results of this analysis lead to a clear view regarding the relevance of more data collection from a predictive performance viewpoint, and a different view from prior systematic learning-curve studies on non-behavioral data.

Before continuing to the details of our benchmarking study, we'd like to point out that data analysis research related to behavior is widespread, and knows many research domains. What we focus on is the use of data on actions/interactions of persons, to make predictions about those persons. This is different from some of the following concepts. Sequential data analysis is a method that allows to examine patterns of behavior over time (Walker, 2016). Behavioral economics studies the effect of psychological processes on economic decisions of individuals. Behavior Informatics is the more general term used for the informatics of behaviors so as to obtain behavior intelligence and behavior insights (Cao, 2010). All these fields are mainly focused on insights that can be gained from or on some behavior (data). We focus on using the behavior data to make predictions on some target variable.

The remainder of this work is organized as follows. In the following section, we present and delimit the data and the classifiers analysed in our comparative study. This results in the analysis of the eleven classifiers shown in Table 2. Section 3 describes the set-up of the

	n	m	PSN	NB	RBF-SVM	LIN-SVM		LR-BGD		LR-SGD		RF	LPR
						L1	L2	L1	L2	L1	L2		
De Cnudde and Martens (2015)	177,761	2,448		**X**			**X**						
Li et al. (2015)	9,489	4,368				**X**							
Goel et al. (2012)	250,000	100,000					**X**						
Junqué de Fortuny et al. (2014)	858,703	108,753	**X**	**X**		X							
Chen et al. (2009)	500,000,000	150,000											**X**
Clark and Provost (2016)	210,004	179,605						X	**X**			X	
Martens et al. (2016)	1,200,000	3,200,000	**X**		X	X							
De Cnudde et al. (2015)	5,000	4,122,418	**X**			**X**							
Yu et al. (2010)	8,407,752	20,216,830					X		**X**				
Stankova et al. (2014)	8,407,752	20,216,830	**X**			X							
Junqué de Fortuny et al. (2013)	8,407,752	20,216,830		**X**									
Agarwal et al. (2014)	2,300,000,000	16,777,216									**X**		
Pandey et al. (2011)	40,000,000	?					**X**						
Perlich et al. (2014)	?	?								**X**	**X**		
Number of wins			**4**	**3**	**0**	**5**		**2**		**2**		**0**	**1**
Total count			4	3	1	9		2		2		1	1

Table 1: Overview of behavioral predictive literature. n is the number of instances, m is the number of features. (Abbreviations of the techniques: PSN = pseudo social network, NB = naive Bayes, RBF-SVM = support vector machine with RBF kernel, LIN-SVM = linear support vector machine, LR-BGD = logistic regression with batch gradient descent, LR-SGD = logistic regression with stochastic gradient descent, RF = random forest, LPR = logistic Poisson regression).

benchmarking study. The results are presented and discussed in Section 4. Finally, we conclude with general remarks and further research avenues in Section 5.

2 Components of the Benchmarking Study

The scope of the benchmarking study is delineated by the type of data analyzed and the classification techniques compared. This section defines and delimits these dimensions and also presents the evaluation procedure.

2.1 Data

We first provide a definition of behavioral data, stating its specific characteristics and comparing it with other high-dimensional and sparse data used in predictive modeling research. Secondly, in order that we understand the degree to which the collection is representative and reproducible, we explain the procedure used to select the collection of behavioral data sets.

2.1.1 Behavioral Data

We follow the definition for big behavioral data given by Shmueli (2016): data originating from human actions and/or interactions. This type of data is special in that it involves human and social aspects such as intention, which is in contrast to data collected from items or products or even physical measurements of people. Being generated from human behavior leads to various differences from non-behavior data (Junque de Fortuny et al., 2013; Shmueli, 2016). This study focuses data sets generated by recording specific, individual behaviors or actions of the people involved, which leads to ultra-high dimensionality and sparseness of the resultant data set, Some modelers instead use a summarization of those features, such as with RFM attributes, capturing behavior along recency, frequency and monetary dimensions (Hu 2005; Hill et al. 2006; De Bock and Van den Poel 2010; Verbeke et al. 2014). Research, however, has shown that better predictive performance is achieved using the most granular form of behavioral data (Clark and Provost 2016; Martens et al. 2016).

The existing literature on predictive modeling from behavioral data provides insight into its main properties (see Table 1). Mostly, this data is characterized by high-dimensionality and sparsity and by having many fine-grained features, where many of them providing additional predictive information—so they are neither uninformative nor completely redundant. Partially due to the sparseness, the observed feature set actually grows as the number of observed instances grows. However, as the number of instances increases, the sparsity grows as well; the average number of features per instance does not grow along with the number of instances or the number of dimensions (Li et al., 2015). This phenomenon has been explained as due to individuals' limited behavioral capital: a person is restricted by resources such as time and money regarding the number

MN-NB	Multinomial naive Bayes
MV-NB	Multivariate naive Bayes
LA-SVM-L2	Least absolute errors support vector machine with a linear kernel and L2 regularization
LS-SVM-L1	Least square errors support vector machine with a linear kernel and L1 regularization
LS-SVM-L2	Least square errors support vector machine with a linear kernel and L2 regularization
PSN	Relational classifier with bigraphs
LR-BGD-L1	Batch gradient descent logistic regression with L1 regularization
LR-BGD-L2	Batch gradient descent logistic regression with L2 regularization
LR-SGD-L1	Stochastic gradient descent logistic regression with L1 regularization
LR-SGD-L2	Stochastic gradient descent logistic regression with L2 regularization
RBF-SVM	Support vector machine with Gaussian kernel

Table 2: The classification techniques studied.

of possible actions she can take (Junqué de Fortuny et al., 2013).

Comparing behavioral data with other high-dimensional data, we find that human behavioral data on the surface resembles text data. The latter is also high-dimensional and sparse with many fine-grained features, many of which contribute to predictive performance (Joachims 1998; Li et al. 2012). Also, text data can be modeled with binary as well as with numeric features, resulting in different performance results (McCallum and Nigam, 1998). Despite the fact that text data consists of many relevant features, most studies in the field of text categorization employ dimensionality reduction (Dumais et al. 1998; Sebastiani 2002; Forman 2003). Textual data is also highly sparse and larger text data sets tend to be sparser; we see this in Figure 1, which plots the data set size and sparsity of all the data sets used in this paper (black dots) as well as all text classification data sets on the UCI Machine Learning Repository (white dots; see Table 3).

Although behavioral data resembles text data in form, there are two important differences. First, the data generating process clearly is different, with behavioral data being generated by human actions and textual data by a language model. The latter has been thoroughly studied statistically and its distribution is governed by laws such as Zipf's law (Zipf, 2016) and Heap's law (Heaps, 1978) among others. Behavioral data, however, is much more complex to capture. One attempt was made in Junque de Fortuny et al. (2013), where destructive human choice behavior was shown to be better modeled by a Wallenius event model than by more traditional models.[1] More research is needed to help us understand the data generating process(es) for behavioral data. A second major difference constitutes the number of features. For all UCI text data sets, the number of feature is substantially lower than for the behavioral data sets analyzed in this study. This makes sense since the number of words in the English language is less than one million, and probably much smaller[2], and the effective vocabulary for particular document classification problems smaller still. In contrast, the number of actions taken in many behavioral settings dwarfs the number of words; see the characteristics of our data below. For example, consider building models from web browsing behavior; the total number of websites currently amounts to $170,712,748$[3], and when considering individual webpages, the number is orders of magnitude larger. Therefore, it makes sense not just to use the surface similarity to conclude that what works for text will work for behavior data, but instead to look draw the conclusion based on a careful analysis of predictive modeling with (this sort of) behavioral data.

[1] Destructive human choice behavior is a specific subclass of human behavioral data where a person's choice to take an action removes that action from the person's future behavior consideration.

[2] Probably less than 250,000; see `https://en.oxforddictionaries.com/explore/how-many-words-are-there-in-the-english-language`.

[3] See `https://news.netcraft.com`.

Data set	n	m	$\bar{m}$	ρ
DBWorld	64	4,702	12,859	95.7269
CNAE-9	1,080	856	7,233	99.2176
NIPS	1,500	12,419	746,316	95.9937
KOS	3,430	6,906	353,160	98.5091
Farm Ads	4,143	54,877	817,141	99.6406
Reuters	8,293	18,933	389,455	99.7520
NIPS87-15	5,811	11,463	4,033,830	93.9442
Newsgroup20	18,774	61,188	2,435,219	99.7880
Enron	39,861	28,102	3,710,420	99.6688
NSF	128,804	25,335	10,449,902	99.6798
NYTimes	300,000	102,660	69,679,427	99.7738
PubMed	8,200,000	141,043	483,450,157	99.9582

Table 3: Data set characteristics of textual classification data sets on UCI Machine Learning Repository.

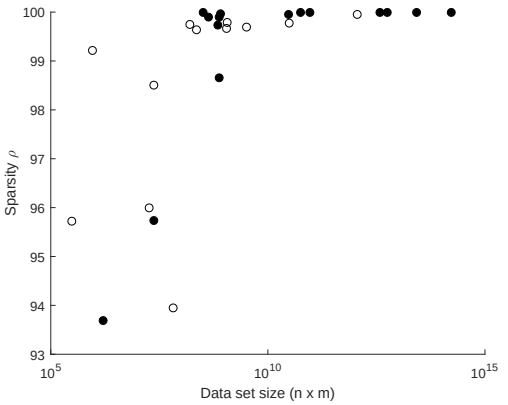

Figure 1: The sparsity (vertical axis) increases along with the size of the data set (horizontal axis). This is demonstrated both for the behavioral data sets analysed in this study (black dots) and the textual data sets from the UCI Machine Learning Repository (white dots).

Next to textual data, one might also observe the similarity (and difference) with datasets used in recommender systems. However our goal and setting are quite different from the recommender one (see Martens et al. (2016)). Firstly, whereas the latter looks at in-domain prediction (for example, using movie ratings to predict other movies' ratings), our setting is out-of-domain prediction (predicting gender based on movie ratings). Secondly, the goal of estimating a relative likelihood of belonging to a class (as done in our classification setting), is different from estimating ratings (as done in recommender systems) in order to find some product to recommend. This makes the tasks both harder and easier for each setting along different dimensions; e.g., recommender systems have to be able to predict fairly well for a massive number of products, but also have the luxury of simply ignoring certain difficult-to-predict products. Finally, the data for the present application simply can be much larger than those in prior work in recommender systems. The Netflix dataset for example, which has received widespread attention, contains 480,189 users and 17,770 movies(Bennett and Lanning, 2007), whereas our datasets have up to 11 million users and 300 million features.

2.1.2 Data Set Selection Procedure

The data set selection in a comparative study may bias the results, implying that care must be taken both in the selection of the data and in the deduction of conclusions regarding learner excellence (Macià et al. 2013; Fernández-Delgado et al. 2014; Macià and Bernadó-Mansilla 2014).

Many studies comparing predictive performance of classifiers use the UCI Machine Learning Repository (Fernández-Delgado et al. 2014; Macià and Bernadó-Mansilla 2014) or an existing benchmark resulting from maturity in a specific research field (Forman, 2003). As predictive research on big behavioral data has only recently emerged, no ready-to-use benchmark is yet present. In order for our results to be generalizable and to put forward an attempt towards a benchmark for future behavioral data research, we looked for behavioral data sets in the various online data repositories listed on the KDnuggets website, which is one of the leading sources of information on data analytics and machine learning.[4] Including publicly available real-world data sets contributes to the reproducibility of this study, and also to the relevance of the results—as the publicly available data sets generally have been drawn from an application that someone cared about, and superior performance on benchmarks is the best empirical path we have for a single study to show results that are likely to translate to other problems (Provost et al., 1998). We enrich the resulting collection with additional real-world behavioral data sets from prior research, but which are not publicly available. These additional data sets do not extend replicability, but are valuable for increasing the sample and the representativeness of the study.

Concretely, the following online data repositories contained relevant data sets: the UCI Machine Learning Repository[5], Yahoo Labs[6], the Stanford Large Network Dataset Collection[7], Kaggle[8], Amazon Web Services data sets[9], the Koblenz network collection (KONECT)[10] and the Max Plank Institute for Software Systems.[11] Some data sets are included in a multi-

[4] See `http://www.kdnuggets.com/datasets/index.html`.
[5] See `http://archive.ics.uci.edu/ml`.
[6] See `http://webscope.sandbox.yahoo.com`.
[7] See `http://snap.stanford.edu/data`.
[8] See `http://www.kaggle.com`.
[9] See `http://aws.amazon.com`.
[10] See `http://konect.uni-koblenz.de`.
[11] See `http://socialnetworks.mpi-sws.org/datasets.html`.

target setting where different targets are predicted in order to use as much data sets as possible.[12]

2.1.3 Data Set Collection

First, a notation is established which will be used throughout this work. A behavioral data set $\mathbf{X}$ consists of n datapoints $\mathbf{x}_i$ with $i = (1, ..., n)$ and $\mathbf{x}_i \in \mathbb{R}^m$. The high-dimensional $\mathbf{x_i}$ represent behavior of an instance i through fine-grained behavioral features j. When modeling behavior in a binary manner, then $x_{i,j} \in \{0, 1\}$. Binary behavior can also be enriched with more detailed information, in that case $x_{i,j} \in \mathbb{N}$. This information might refer to frequency (for example in the case of visiting behavior) or preference (for example in the case of rating data). In this classification setting, Y models the target variable that should be predicted and is a vector of size n with $y_i \in \{-1, +1\}$.

In total, 41 behavioral data sets are used, originating from 15 real-world problems. The *MovieLens* data set[13] contains movie-rating data from users. Based on these ratings, predictions are made concerning the gender and age of a user. Two versions are available: one with 100,000 features and one with 1,000,000 features. The latter is also used to predict the genre of the movies based on users' ratings. Eighteen data sets are constructed in order to translate this multi-class problem to a binary problem. Yahoo Labs[14] makes available the *YahooMovies* data set which contains movie-rating data, analogous to the *MovieLens* data set. Here, also the gender and age of the users are predicted. The *Ecommerce* data set originates from the PAKDD2015 challenge with the goal of predicting gender based on product viewing data on an e-commerce website[15]. Next, the *TaFeng* data set contains shopping transactions of users and the goal is to predict the users' age (Huang et al., 2005). In the *BookCrossing* data set, books are rated by members of the BookCrossing community and based on these ratings, the age of the user is predicted (Ziegler et al., 2005). The *LibimSeTi* data set contains ratings of dating profiles by users of the dating service LibimSeti (Brozovsky and Petricek, 2007). Based on these profile ratings, the gender of the user is inferred. The KDD cup 2015 challenge aspires to predict the MOOC dropout rate from the online learning platform XuetangX based on prior online course behavior. The *A-Card* data sets consist of user-visiting behavior from a city loyalty card on which three predictions are made (De Cnudde and Martens, 2015). First, cashout prediction consists of predicting whether a user will trade collected points for a benefit. Second, an assertion is made with respect to the user becoming inactive which is referred to as defect prediction. Third, for each user and five locations, a prediction is made whether that location will be visited in the near future. The *Fraud* data set consists of transactional information concerning payments between Belgian and foreign companies and attempts to predict whether a company is involved in fraudulent activities (Junqué de Fortuny et al., 2014). In Martens et al. (2016), the *Banking* data set is constructed by collecting debit transactions from customers of a bank. With this payment data, a prediction is made concerning the possible purchase of a financial product offered by the bank. The goal of the *KDDa* data set (Yu et al., 2010) from the 2010 KDD cup challenge is to predict the performance of students on an algebraic test based on their past performance. In the *Flickr* data set, the transactions consist of users tagging pictures as being their 'favorite' and we predict the number of comments a picture has (Cha et al., 2009). For the proprietary *Car* data set, predictions regarding the interest in a car advertisement are made based on users' web visiting behavior.

Table 4 summarizes some general characteristics related to the data sets. Judging from this summary, a great variety of data sets is present in terms of size (both in the instance as well as in the feature dimension), the nature of the predictive variable, the n-m relation ($n \ll m$, $n \gg m$ and $n \approx m$) and the balance b. Since these are real-life data sets, in most cases, the distribution of the classes is unbalanced (Yang and Wu, 2006). For the *fraud* data set, the highest imbalance is achieved, as the number of fraudulent organizations in comparison to the number of non-fraudulent organizations is very low (Liu et al., 2007).

[12] These multi-target problems are appropriately handled in subsequent statistical comparisons.

[13] See `http://grouplens.org`.

[14] See `http://webscope.sandbox.yahoo.com`.

[15] See `https://knowledgepit.fedcsis.org`.

Data set	Target variable	Binary	Numeric	n	m	$\bar{m}$	ρ	b
MovieLens100k	age	✓	✓	943	1,682	100,000	93.6953%	42.31
MovieLens100k	gender	✓	✓	943	1,682	100,000	93.6953%	28.95
MovieLens1m	age	✓	✓	6,040	3,883	1,000,209	95.7353%	43.36
MovieLens1m	gender	✓	✓	6,040	3,883	1,000,209	95.7353%	28.29
Yahoo Movies	age	✓	✓	7,642	106,363	221,330	99.9727%	21.09
Yahoo Movies	gender	✓	✓	7,642	106,363	221,330	99.9727%	28.87
MovieLens10m	action	✓	✓	10,681	69,878	10,000,053	98.6602%	13.79
MovieLens10m	adventure	✓	✓	10,681	69,878	10,000,053	98.6602%	5.36
MovieLens10m	animation	✓	✓	10,681	69,878	10,000,053	98.6602%	1.51
MovieLens10m	children	✓	✓	10,681	69,878	10,000,053	98.6602%	1.75
MovieLens10m	comedy	✓	✓	10,681	69,878	10,000,053	98.6602%	28.29
MovieLens10m	crime	✓	✓	10,681	69,878	10,000,053	98.6602%	5.50
MovieLens10m	documentary	✓	✓	10,681	69,878	10,000,053	98.6602%	4.09
MovieLens10m	drama	✓	✓	10,681	69,878	10,000,053	98.6602%	29.57
MovieLens10m	fantasy	✓	✓	10,681	69,878	10,000,053	98.6602%	0.43
MovieLens10m	film noir	✓	✓	10,681	69,878	10,000,053	98.6602%	0.24
MovieLens10m	horror	✓	✓	10,681	69,878	10,000,053	98.6602%	5.10
MovieLens10m	musical	✓	✓	10,681	69,878	10,000,053	98.6602%	0.41
MovieLens10m	mystery	✓	✓	10,681	69,878	10,000,053	98.6602%	0.43
MovieLens10m	romance	✓	✓	10,681	69,878	10,000,053	98.6602%	0.56
MovieLens10m	sci-fi	✓	✓	10,681	69,878	10,000,053	98.6602%	0.66
MovieLens10m	thriller	✓	✓	10,681	69,878	10,000,053	98.6602%	1.23
MovieLens10m	war	✓	✓	10,681	69,878	10,000,053	98.6602%	0.19
MovieLens10m	western	✓	✓	10,681	69,878	10,000,053	98.6602%	0.86
Ecommerce	gender	✓		15,000	21,880	33,455	99.9898%	21.98
TaFeng	age	✓		31,640	23,719	723,449	99.9036%	39.67
BookCrossing	age	✓	✓	167,175	337,921	838,364	99.9985%	29.04
LibimSeTi	gender	✓	✓	137,806	220,970	15,656,500	99.9486%	44.53
KDD2015	MOOC dropout	✓	✓	120,542	5,891	1,919,150	99,7300%	20.71
A-Card	cashout	✓	✓	177,761	2,448	435,244	99.9000%	6.71
A-Card	defect	✓	✓	177,761	2,448	435,244	99.9000%	13.20
A-Card	Permeke	✓	✓	177,761	2,448	435,244	99.9000%	7.29
A-Card	Wezenberg	✓	✓	177,761	2,448	435,244	99.9000%	2.18
A-Card	MAS	✓	✓	177,761	2,448	435,244	99.9000%	1.82
A-Card	Roma	✓	✓	177,761	2,448	435,244	99.9000%	0.96
A-Card	Zoo	✓	✓	177,761	2,448	435,244	99.9000%	0.85
Fraud	fraudulent	✓		858,131	107,345	1,955,912	99.9979%	0.0064
Banking	interest in product	✓		1,204,726	3,192,554	20,914,516	99.9995%	0.35
KDDa	task performance	✓		8,407,752	20,216,830	305,613,510	99.9998%	14.70
Car	interest in ad	✓		9,108,905	2,936,810	65,464,708	99.9998%	0.70
Flickr	comments	✓		11,195,144	497,472	34,645,469	99.9994%	27.05

Table 4: General characteristics concerning the data sets (ordered by ascending n): the target variable being predicted, whether binary and numeric versions are available, the number of instances n, the number of features m, the number of active elements $\bar{m}$, the sparsity ρ defined as $\rho = 1 - (\bar{m}/(n \times m))$ and the balance b (percentage of positive instances in the target variable).

As mentioned and as demonstrated in Table 4, the sparsity ρ of behavioral data sets is extreme due to limited behavioral capital (Junqué de Fortuny et al., 2013). Figures 2-3 show the probability distributions of the number of features per instance (Figure 2) and the number of instances per feature (Figure 3) which we refer to here as the sparsity distributions. These distributions provide support for the limited behavioral capital explanation. It is clear from the sparsity distributions for the instances (Figure 2) that most instances have a very low number of active (non-zero) features. From the tail of the distributions, it can be observed that instances with a large number of active features are much less frequent. Conversely, looking at the sparsity distributions of the features in Figure 3, also the probability of a feature being present in many instances' behaviors is low. This makes sense when looking at the *Banking* data set for example: the majority of users have payment transactions only with a small fraction of all possible payment receivers. Also, the majority of the payment receivers have payment relations with only a small fraction of all clients of the bank.

From the sparsity distributions for the number of features per instance, an additional difference between behavioral and textual data becomes clear. Figure 4 shows this sparsity distribution for the *Newsgroup20* text data set[16]. For text data, this inverted U shape is based at least in part on the relationship between documents and sentences, and thus the sentence-length distribution (Sigurd et al., 2004); it clearly

[16] We show the sparsity distribution for the number of features per instance for one textual data set only. For the other textual data sets from the UCI Machine Learning Repository, we generally find similar shapes.

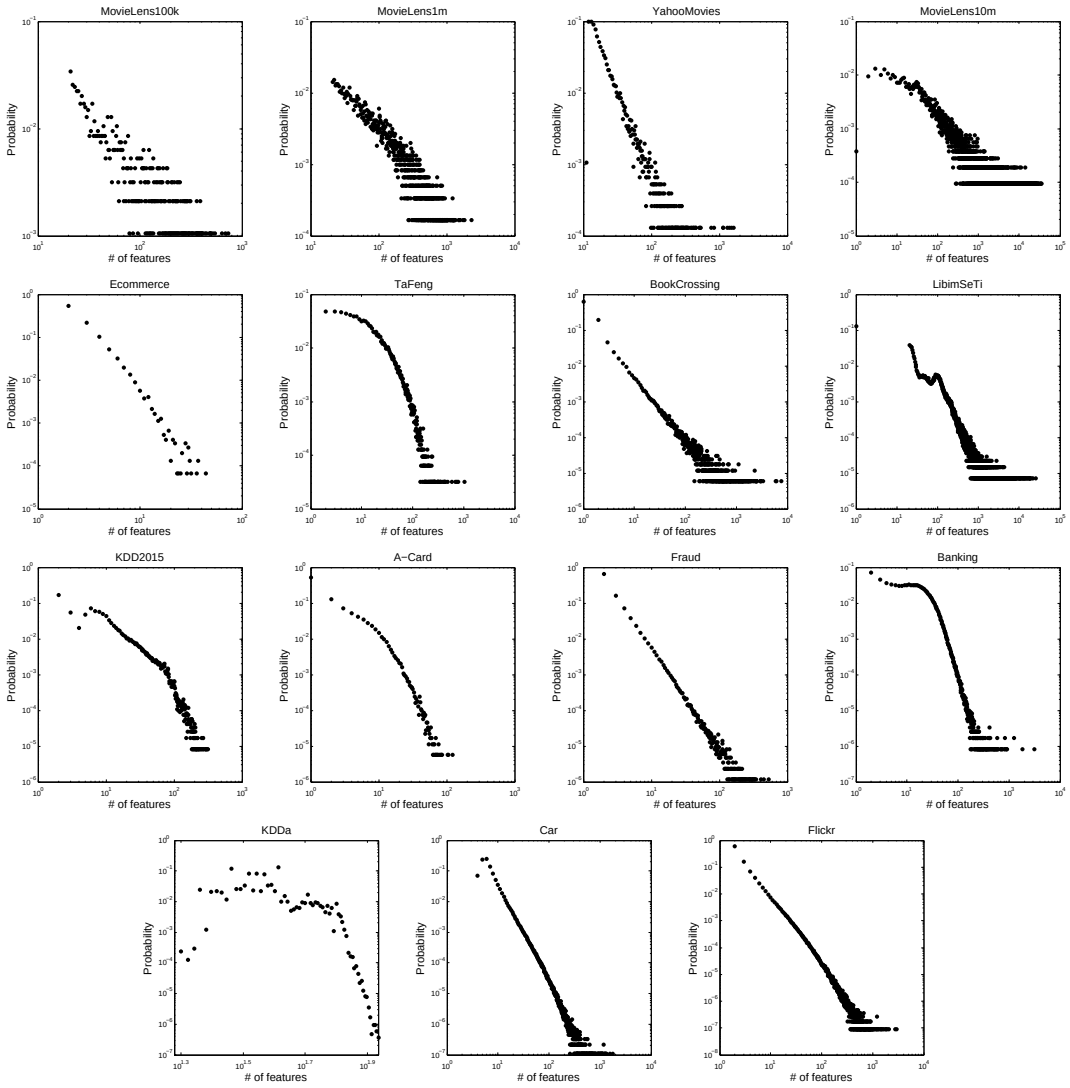

Figure 2: Sparsity distributions for the number of features per instance.

differs from almost all of the distributions for the behavioral data sets (in Figure 2).

2.1.4 Classification Techniques

As stated in Section 1, the goal of our study is to provide insight and guidance to researchers and practitioners when faced with large, behavioral data. In order to study a relevant and representative selection of classifiers (Macià and Bernadó-Mansilla, 2014), we take the following approach. We examine the existing literature performing predictive analyses on behavioral data to determine which techniques have been used and why, and what problems were encountered during the analysis. We focus on literature specifically analyzing fine-grained, high-dimensional, and sparse behavioral data and summarize the employed classification techniques in Table 1 (marked with an 'X' in the appropriate column). While constructing the table, the following rules were applied. If no explicit mention is made of the number of instances n and/or the number of features m, we denote this with a question mark. In case a paper analyzes several data sets, the largest is shown. For each data set, a bold 'X' represents the best-performing technique. At the bottom, the table shows the total number of occurrences of each technique, along with the number of times it performed best among the techniques used (also shown in bold). Note that when only one technique is analyzed in a paper, it is nonetheless denoted in boldface.

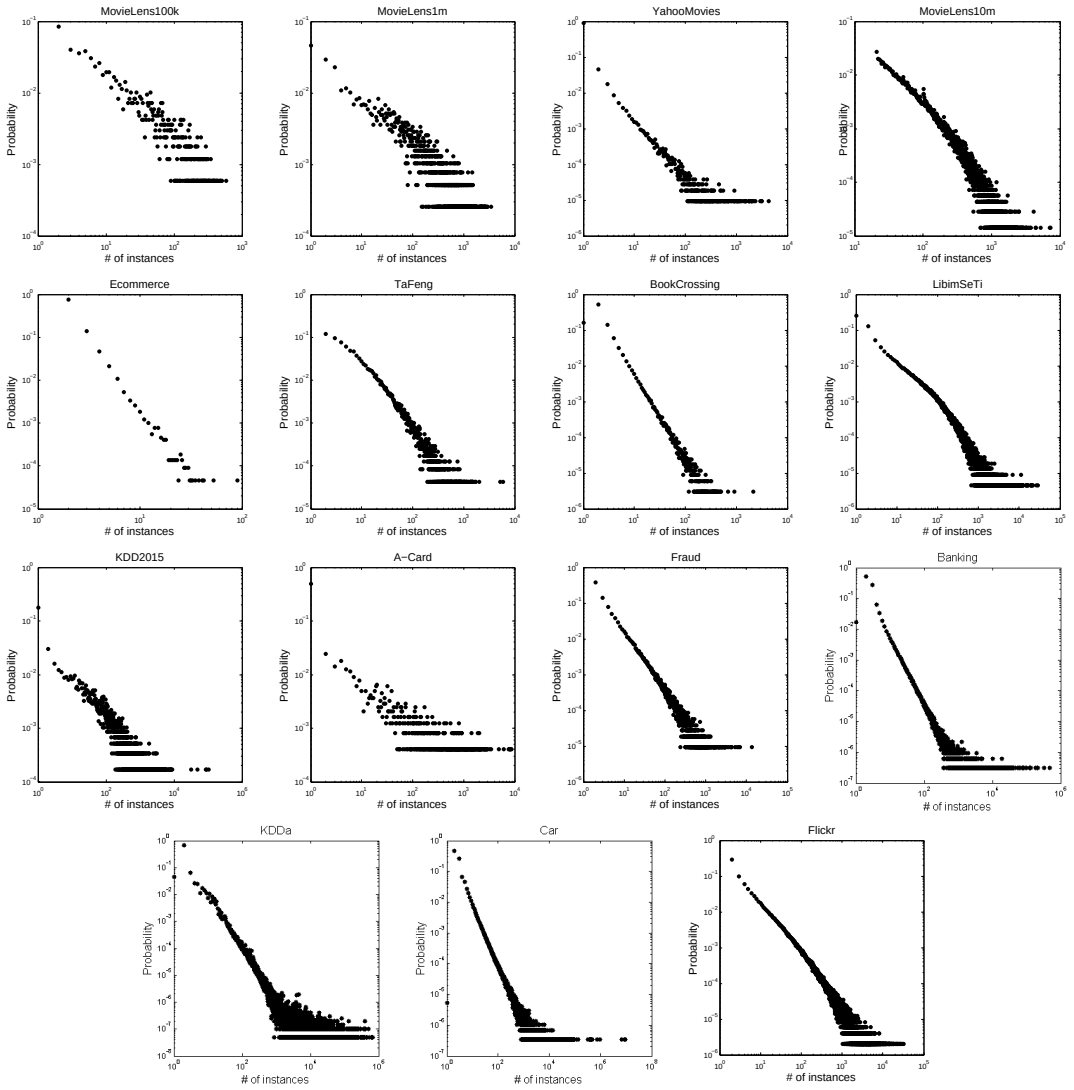

Figure 3: Sparsity distributions for the number of instances per feature.

For linear support vector machines and logistic regression, the type of regularization is indicated. When the authors did not specify which type was used, an 'X' is put in the middle. From Table 1, it is observed that some consensus seems to exist in prior work regarding what method to use: linear SVMs are most frequently used, along with L2 regularization. The papers specifically mention the use of linear SVMs as fast and adequate in very high-dimensional contexts. For naive Bayes, many papers mention its speed and performance on textual data as justifications. Logistic regression, interestingly, performs at least as well or better than the linear SVMs in all reported comparisons. Lastly, there is the SVM with radial basis function kernel, capable of finding non-linear patterns. Most papers, however, condemn this technique for its lack of scalability. We also examine the classifiers used in text classification research and find that mostly support vector machines, naive Bayes, random forests and nearest neighbor classifiers are used when the analysis is performed without dimensionality reduction (Joachims 1998; Colas and Brazdil 2006).

The final selection of classifiers for our comparative setting is naturally restricted by the characteristics of the data, which impose specific challenges. Random forests, for example, are not adept at handling massively high dimensionality with many relevant features (Do et al., 2009): complex interactions between the features are ignored due to the division of the

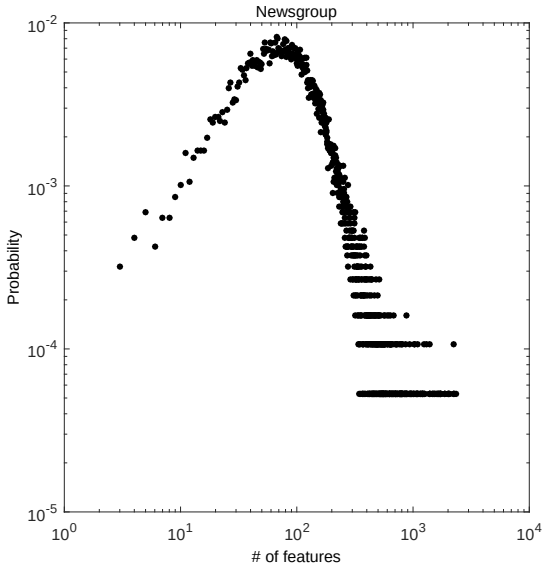

Figure 4: Sparsity distribution for the number of features (words) per instance (document) for the *Newsgroup20* data set.

training space in mutually exclusive subspaces. Furthermore, it is simply infeasible to build trees on some of the massive feature spaces and still get non-trivial classifications on the ultra-sparse instances. When using nearest neighbors, the massive dimensionality drastically magnifies the neighborhood search space, which impedes the search for similar data points and significantly increases the run time. (As noted above, text classification problems do not reach the same massive dimensionality as behavioral data problems.) Thus, despite their use for text classification, we do not select nearest neighbor or random forests for inclusion in the paper's comparison.

The final selection of classifiers listed in Table 2. Regarding the variations with respect to loss function and regularization for support vector machines, we employ the most commonly used options, which are those offered by the widely-used LIBLINEAR package (Fan et al., 2008). Note that for PSN, a Python version is publicly available[17]. In Appendix A, details of the classification techniques are given, along with information regarding implementation and computational complexity.

[17] `https://github.com/SPraet/SW-transformation/`

2.2 Performance Measures

2.2.1 Area under ROC-curve (AUC)

Accuracy is a fairly intuitive and often-used measure of performance (King et al. 1995; Lim et al. 2000): it expresses the percentage of correctly predicted instances (Fawcett, 2006). However, it is influenced by class imbalance. Since the bulk of the data sets in this study come from real-life classification tasks and exhibit class imbalance (Table 4), accuracy is not a satisfactory measure (Provost et al., 1998).

Instead we use the the Area Under ROC-Curve (AUC). ROC (Receiver Operating Characteristic) space is used to plot the performance of classifiers in terms of the true positive rate (TP) and the false positive rate (FP), on the Y-axis and the X-axis respectively. This is done by ranking the classifier's prediction scores for data points in the test set in a descending fashion while iteratively lowering the threshold for classifying an instance as positive. The AUC value is a summarizing scalar representing the area under this performance curve (Fawcett, 2006). Thus, it expresses the models' ability to rank instances in a descending fashion in terms of their prediction score or, in other words, the probability of a classifier to rank a randomly chosen positive instance higher than a randomly chosen negative

instance. We scale the AUC to [0,100], and so an AUC of 50 corresponds to a model performing no better than random guessing. A perfect model has an AUC of 100.

2.2.2 Statistical Significance Test

Two statistical tests are used in order to elect an algorithm or a group of algorithms as better or best performing: the Wilcoxon signed rank test and the Friedman test, both proposed by Demšar (2006). The former compares two treatments of a collection of data sets (used to contrast binary versus numeric data); the latter is used to compare a collection of treatments (for comparing all classifiers).

The Wilcoxon signed-rank test is a non-parametric test which first computes the absolute differences in performance between two treatments of a collection of data sets. These differences are ranked and summarized in two variables $R+$ and $R-$ representing the sum of ranks where the second treatment, respectively the first treatment, performs better. The lowest value $T = \min(R+, R-)$ is compared to a Wilcoxon critical value. If T is equal to or lower than this value, the null hypothesis stating that the two treatments perform equal can be rejected and a significant difference is found.

In the Friedman test for each data set separately, the performance values for each method are ranked. The average rank $AR_j = \frac{1}{N}\sum_i r_i^j$ of each algorithm is calculated, with r_i^j the rank of the j-th algorithm on the i-th data set. The Friedman statistic is defined as

$$\chi_F^2 = \frac{12N}{K(K+1)}\sum_j AR_j^2 - \frac{K(K+1)^2}{4},$$

with N the number of data sets and K the number of algorithms. Iman and Davenport (1980) state that this χ^2 approximation results in an overly conservative statistic with too small a critical region and present an updated approximation

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)-\chi_F^2},$$

distributed according to an F-distribution with $(K-1)$ and $(K-1)(N-1)$ degrees of freedom. This value is compared to a critical value corresponding to an F-distribution and a significance level α, resulting in either accepting or rejecting the null-hypothesis that all algorithms are equivalent. In the latter case the Nemenyi post-hoc test is performed. This test defines a critical difference

$$CD = q_\alpha\sqrt{\frac{K(K+1)}{6N}},$$

with q_α a critical value based on the Studentized range statistic divided by $\sqrt{2}$. Two classifiers demonstrate significantly different performance measures if their average ranks differ more than the critical difference value.

2.2.3 Learning Curves

Learning curves show performance variations of learning algorithms as a function of the size of the training set. The goal is to get insight into performance generalization of the algorithm regarding data set size (Perlich et al., 2003). The performance values in terms of AUC (which assume independence of data size) now gain a substantial level of detail as performance is compared over different techniques *and* over different data set sizes (Fernández-Delgado et al. 2014; Macià and Bernadó-Mansilla 2014).

Concretely, a learning curve plots the performance as a function of the training set size, generally on a logarithmic scale. AUC is used as a performance measure and the training set size is varied separately in the dimension of the instances and the features. For the instance dimension, increasing samples are drawn (uniformly at random) from the original training data. For the feature dimension, learning curves are built in two ways. First, increasing samples of the features are drawn uniformly at random. These learning curves are built to assess performance variations over the number of features, regardless of their predictive value. Secondly, the information value of each feature is determined. Learning curves are then built by taking increasing feature samples according to their descending information value. This approach enables us to relate performance variations to the importance of the features, and to assess the relevance of many fine-grained features in predictive performance. The information value of a feature can be assessed by a plethora of metrics (Forman 2003; Guyon and Elisseeff 2003). We employ the information gain metric here as it is a fairly quick and accurate way to determine value in separate features (Forman 2003). Information gain

models the reduction of entropy in the predictive variable brought along by the presence of a feature f

$$\text{Information Gain}(f) = H(Y) - H(Y|f),$$

with Y the classification values of a training set, f a feature and $H(Y)$ the entropy of Y.

3 Experimental Set-Up

Before training the model, a third of the training set (sampled uniformly at random) is set aside as validation set for parameter selection. Model selection with grid search is performed to find an optimal value for the regularization parameter C (for logistic regression and the SVMs) and the kernel parameter γ for RBF-SVM. An initial grid ($[2^{-5}, 2^{-3}, \ldots, 2^{15}]$ for C and $[2^{-15}, 2^{-13}, \ldots, 2^{3}]$ for γ) (Hsu et al., 2003) is explored, that is, models are constructed and tested on the validation set with the grid parameter values. Based on the best performing model, a new grid is built around the best value. These grids are then iteratively improved (each time building a more fine-grained grid around the best value found in the previous iteration, up to three times) and the best resulting value is finally used in building the classification model on the training set.

Since multivariate naive Bayes and PSN expect binary data (see Appendix A), the numeric information is modeled in a binary fashion through unary encoding[18]. This so-called thermometer code translates non-negative, numeric feature values $x_{i,j}$ with maximum range R into $x_{i,j}$ ones followed by $(R - x_{i,j})$ zeros. In theory, this increases the dependency between the features and violates the naive Bayes assumption. However, the approach has been shown to result in good predictions even with dependent features (Hand and Yu, 2001). This unary expansion results in higher-dimensional data for the numeric analysis of both PSN and MV-NB.

Dependence of the results on data sampling is a relevant issue in benchmark study design, impacting the reliability of the study (Fernández-Delgado et al. 2014; Macià and Bernadó-Mansilla 2014). Therefore, k-fold cross validation is used which determines k disjoint partitions through sampling uniformly at random from the entire data set, and using each partition as test set and the remaining $k-1$ as training set. Commonly, k is set to 10 which has been shown sufficient in reducing bias and variance (Kohavi, 1995). The folds are equal for all classifier executions, such that sound comparisons across classifiers can be made. Moreover, this fulfills the necessary conditions of stable results as mentioned in Demšar (2006); the statistical tests demand 'reliable estimates of the classifier's performance'.

The learning curves are built as follows:

1. For the learning curves in the instance dimension: For each of the ten cross-validation folds, repeatedly take random subsamples of the training set with an increasing n_l-value ($n_l \in 1, ..., n$).
2. For the learning curves in the feature dimension (random features): For each of the ten cross-validation folds, repeatedly take random subsamples of the training set with an increasing m_l-value ($m_l \in 1, ..., m$). Adjust the corresponding test set according to the selected features.
3. For the learning curves in the feature dimension (feature selection): For each of the ten cross-validation folds, repeatedly take subsamples of the training set according to descending information value of the features with increasing m_l-value ($m_l \in 1, ..., m$). Adjust the corresponding test set according to the selected features.

The analyses were performed on an Intel i7 processor with 4 physical cores, 3.40 GHZ clock rate and 16 GB RAM.

Among the investigated techniques, RBF-SVM stands out with its $\mathcal{O}(n^2)$-$\mathcal{O}(n^3)$ complexity as mentioned in Appendix A. This clearly is not scalable with respect to the sizes of many of these data sets. Therefore, for the largest dimensions (starting from *BookCrossing* in Table 4), the entire data set could not be used when comparing AUC and time performance. A random subsample of size 2^{15} is used as a proxy for these data sets.

[18] Converting numeric to binary features can also be done through dummy encoding. Both dummy encoding and unary encoding gave similar results and we discuss the unary approach in more detail here.

4 Experimental Results

We now present the results of the comparison of the methods, focusing on those results that will be most useful to inform the choices of future researchers and practitioners. We first discuss the results which are independent of training size variations, followed by the analysis of the learning curves.

4.1 Performance Analysis

The analysis of performances is divided into the following parts: (1) comparison of classification and time performance, (2) comparison of the effect of binary versus numeric data on AUC and time, and (3) interpretation of the performance results.

4.1.1 Comparison of Classification and Time Performance

Table 5 and Table 6 report the AUC values for all binary and numeric data sets, respectively (ordered by ascending maximum AUC value). For each data set, the best AUC is denoted in boldface. Also, the average rank per technique is shown where for each data set rank 1 is given to the best technique and rank 10 is given to the worst performing technique. For each algorithm, also the number of times it performs best is given. The results for RBF-SVM are pictured somewhat isolated because the technique is not always run on the entire data set.

For the binary data sets in Table 5, LR-BGD-L2 and PSN perform best. MV-NB performs better than MN-NB. The techniques optimized with L2 regularization (LA-SVM-L2, LS-SVM-L2, LR-BGD-L2 and LR-SGD-L2) have better performance compared to their counterparts with L1 regularization. The SGD variants of logistic regression demonstrate worse performance than BGD. Regarding RBF-SVM, we observe that even when using a sample for the larger data sets, it only performs the worst in a minority of cases; however, it almost never is the best.

In line with the no-free-lunch theorem (Wolpert, 1996), linking classifier performance to underlying data characteristics is essential and reveals their specific domain of competence (Macià et al., 2013). This creates the need for meta-analyses providing more detailed insight. Figure 5 shows a decision tree denoting which classifier performs best dependent on extrinsic data characteristics. These consist of instance dimension n, feature dimension m, number of active elements $\bar{m}$, sparsity ρ, balance b and nature of behavior (rating, location, transactional, interest). Note that for this tree, only one of the eighteen *MovieLens_genre* data sets (and only one of the *MovieLens100k*, *MovieLens1m*, *A-Card* and *YahooMovies* data sets) is used to train the tree to reduce overfitting. The tree conceptualizes the following findings:

- For small, imbalanced data sets, the PSN approach leads to higher classification performance (for example *MovieLens_scifi*, *YahooMovies_age*).
- For large, imbalanced data sets, MV-NB leads to higher classification performance (for example *A-Card_defect*, *Flickr*, *Fraud*, *Banking*).
- For very large, imbalanced data sets, LR-BGD-L2 leads to higher classification performance (for example *kdda*, *Car*).
- For balanced data sets, LR-BGD-L2 leads to higher classification performance (for example *MovieLens100k_age*, *MovieLens1m_gender*, *BookCrossing*).

The findings from the decision tree are emphasized by an additional logistic regression performed for each technique on the data set characteristics and whether that technique performs best (1) or not (0). Significant regression coefficients were found for MV-NB (instance dimension n, balance b) and for LR-BGD-L2 (balance b). Note that both these meta-analyses are purely to understand where the different techniques are performing better or worse in this study—not to present generalizable results.

Turning to the numeric data sets, Table 6 shows that PSN performs best, followed by LR-BGD-L2. Multivariate naive Bayes and LR-SGD perform worst. In contrast to binary data sets, MN-NB overall has lower rank than MV-NB. Analogous to the binary data sets, L2 regularization and BGD perform better than L1 regularization and SGD respectively. RBF-SVM has a better score compared to the one for binary data sets; however, it does not perform among the best techniques despite its capability of capturing complex relations (Chang and Lin, 2011). No decision tree was built here

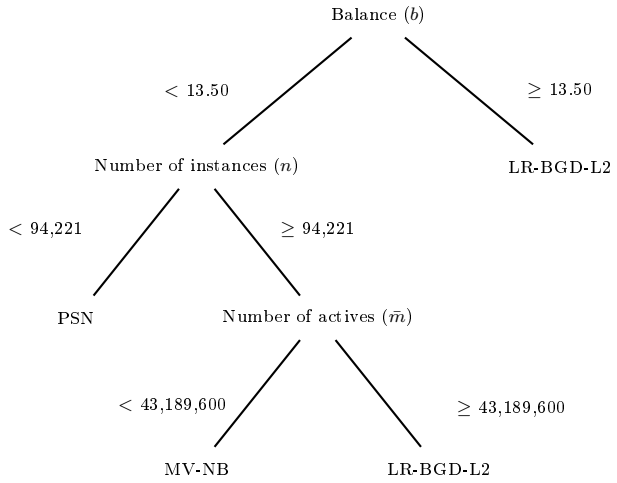

Figure 5: Decision tree visualizing the most discriminative data set characteristics for all classifiers for **binary** data sets.

due to only having a sample of 8 representative numeric data sets.

In order to extrapolate these findings to a larger population of behavioral data sets, ideally the data sets should form a random and representative sample of the population. However, the data collection consists of multiple multi-target problems (*MovieLens*, *YahooMovies* and *A-Card*). These are subsets of prediction problems with the same features but different targets. (Encouragingly, there is remarkable consistency in their best-performing techniques). Two approaches were taken to address this.

First, we randomly selected one data set in each multi-target problem to represent the others. Second, we weighted the ranks of these multi-target data sets in order for all information to be present in the analysis. Both approaches lead to the same statistical conclusions, and in what follows we present the former. Concretely, the statistical test is performed with a sample size of 15 for the binary data and 8 for the numeric data.

The non-parametric Friedman test is performed at a $\alpha = 0.05$ significance level. Following the graphical representation proposed by Demšar (2006), Figure 6 sets out the average ranks of the classification techniques both for AUC (dashed lines) and execution time (solid lines), for the binary data sets (top) and for the numeric ones (bottom). Horizontal connections between techniques denote groups of algorithms which show no significant performance differences. Note that the ranks in Figure 6 differ from the ranks in Table 5 and Table 6 since a different sample is used.

From Figure 6 (binary, top), we observe that LR-BGD-L2 performs better than MN-NB and RBF-SVM. Although LR-BGD-L2 has the best classification performance, it is very slow in terms of run time.[19] In contrast, MN-NB is quite fast, but unfortunately it performs quite poorly in terms of AUC. RBF-SVM achieves the worst AUC and is the slowest. The best performing method with respect to time is PSN. Overall, PSN achieves a very respectable AUC-time trade-off.

From Figure 6 (numeric, bottom), we cannot distinguish the techniques statistically in terms of AUC. The small sample size of 8 is the main reason for this. Here, also, PSN and MN-NB are the fastest and the non-linear RBF-SVM the slowest. Also, logistic regression and the L2-regularized techniques are very time-consuming.

Figure 7 presents the Pareto front for both types of behavioral data, clearly demonstrating the multi-objective trade-off between AUC and time: if more computational resources are available (further right), better classification predictions are reached. Note that the majority of techniques on the Pareto fronts use L2 regularization.

[19] For the *kdda* data set for example, LR-BGD-L2 find a solution in 2.5 hours, while PSN finds one in only 3.5 seconds.

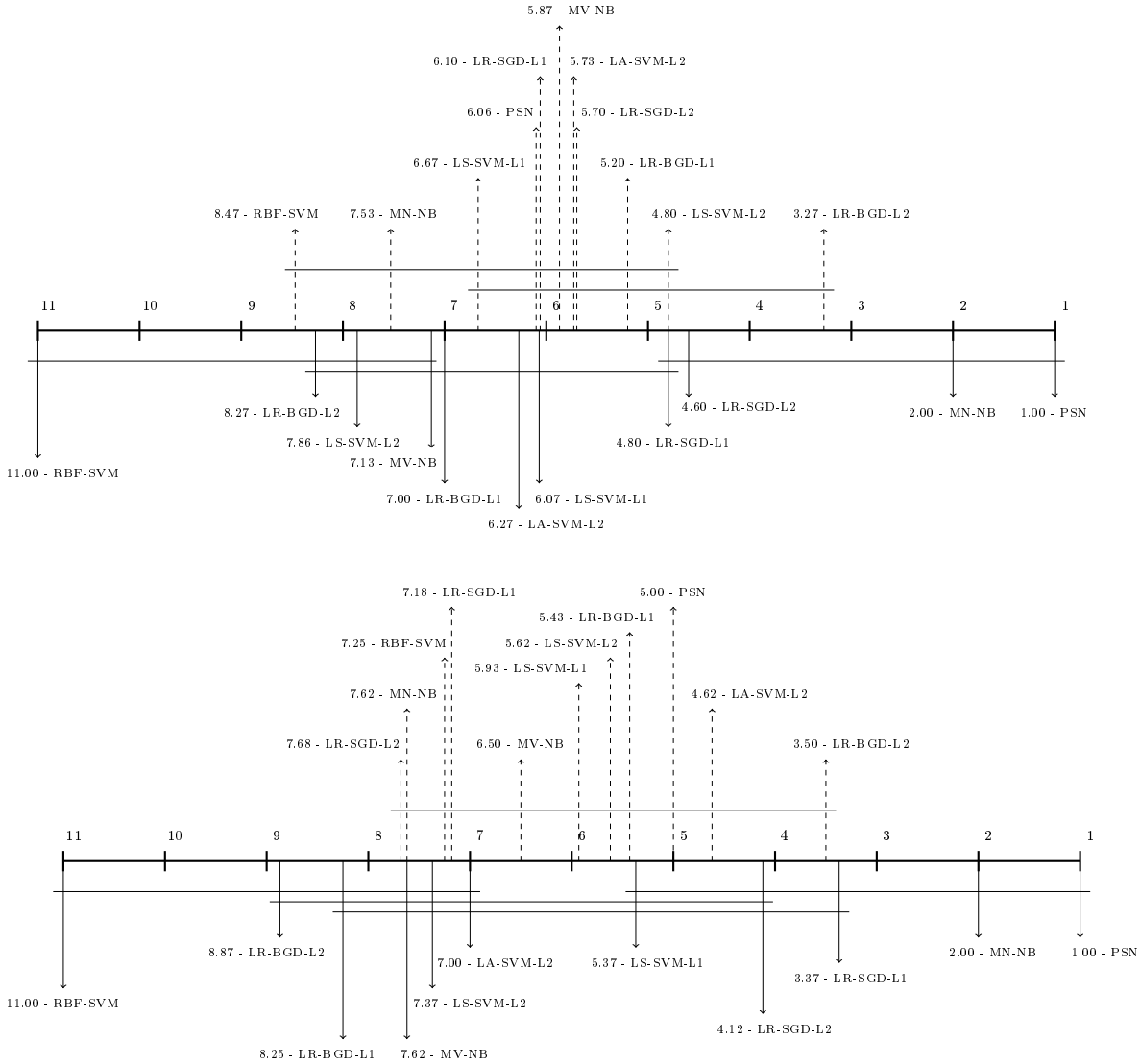

Figure 6: Statistical significant differences (at $alpha = 0.05$ significance level) between the methods in terms of AUC (dashed lines) and time (solid lines) for a random sample of binary (top) and numeric (bottom) data sets. The horizontal lines depict a group of methods for which no significant difference was found.

	MN-NB	MV-NB	LA-SVM-L2	LS-SVM-L1	LS-SVM-L2	PSN	LR-BGD-L1	LR-BGD-L2	LR-SGD-L1	LR-SGD-L2	RBF-SVM
BookCrossing	56.17	56.05	56.16	54.46	56.24	53.54	54.78	**56.25**	55.10	55.13	53.48
YahooMovies_age	59.55	65.20	63.06	63.82	64.57	65.28	63.54	64.84	61.52	61.55	**71.94**
banking	54.79	**68.17**	54.25	53.40	54.62	67.08	53.73	54.95	66.28	67.24	53.45
TaFeng	**71.92**	70.58	70.57	69.52	71.19	69.86	69.80	71.36	65.77	66.05	63.04
MovieLens_crime	74.59	72.30	75.09	75.07	75.83	**76.33**	75.16	76.13	72.52	73.07	68.34
A-Card_Goto_MAS	63.73	**76.50**	68.58	63.36	63.27	75.51	64.93	64.97	75.55	76.50	64.78
MovieLens_adventure	73.6	59.72	73.52	73.79	73.94	**76.57**	73.79	73.98	68.86	67.64	70.07
fraud	76.27	**77.16**	57.13	50.41	55.68	76.87	52.66	55.48	72.75	77.11	69.09
MovieLens_100k_gender	74.48	69.63	74.73	73.02	77.09	74.71	72.82	**77.21**	73.78	74.49	75.75
Car	57.71	77.52	69.98	77.14	74.21	71.06	75.53	**77.70**	72.31	72.09	55.83
MovieLens_fantasy	61.99	56.13	61.35	60.43	62.27	**78.3**	61.23	62.22	61.00	61.80	69.85
MovieLens_romance	65.43	73.06	70.16	70.37	69.44	**79.56**	71.25	68.20	69.31	69.28	61.81
Ecommerce	77.24	55.85	79.54	75.31	79.60	68.30	76.02	**79.62**	79.26	79.26	71.95
YahooMovies_gender	73.81	79.30	79.92	79.08	80.30	80.18	79.14	**80.43**	75.36	75.44	78.38
MovieLens_mystery	59.86	57.32	69.61	69.85	69.70	**81.07**	70.32	68.03	61.95	62.80	69.26
A-Card_Goto_Permeke	77.47	80.84	61.13	64.5	64.34	78.27	63.14	63.14	**82.35**	**82.35**	81.86
MovieLens_children	80.97	73.02	79.10	79.78	79.54	**83.7**	79.83	79.7	76.07	75.05	80.75
MovieLens_drama	73.65	65.60	82.18	82.76	83.27	76.52	82.84	**83.74**	79.35	79.17	70.07
MovieLens_thriller	70.00	73.27	74.53	75.16	75.03	**84.04**	76.75	75.08	72.97	73.45	70.31
kdd2015	70.79	81.14	77.95	79.19	79.25	83.73	79.84	79.89	**84.43**	83.71	81.57
kdda	78.91	78.32	81.17	84.44	83.92	79.82	84.33	**85.50**	82.28	78.88	70.15
MovieLens_1m_gender	80.54	76.84	84.05	83.83	84.83	81.33	84.17	**85.20**	80.55	80.56	82.57
A-Card_Goto_Wezenberg	79.74	84.57	53.25	56.88	57.00	84.92	55.19	55.21	85.36	**85.40**	79.37
A-Card_defect	51.41	**85.50**	76.53	75.49	75.14	78.68	70.93	75.39	81.68	81.72	65.65
flickr	77.77	**85.99**	73.48	76.77	76.22	76.63	77.08	76.97	84.38	84.25	80.15
MovieLens_comedy	77.02	77.82	84.45	85.03	85.84	78.21	85.29	**86.13**	82.19	82.09	74.82
A-Card_Goto_Roma	70.78	**86.60**	60.12	56.52	56.59	86.32	58.03	57.96	85.37	85.38	67.79
MovieLens_action	82.07	66.30	85.90	86.48	86.53	82.52	86.76	**86.89**	83.53	83.37	84.33
A-Card_Goto_Zoo	71.71	86.58	60.34	55.73	55.70	**87.05**	57.71	57.64	85.30	85.74	72.56
MovieLens_100k_age	79.43	77.20	87.09	84.17	87.71	80.21	85.08	**87.95**	84.10	83.33	81.61
MovieLens_animation	84.83	70.00	87.29	87.53	87.16	85.91	**88.04**	87.10	80.75	80.55	84.86
MovieLens_scifi	76.73	68.08	79.08	78.75	78.85	**88.50**	80.42	79.66	73.54	73.94	69.64
MovieLens_documentary	83.55	71.40	87.94	87.45	88.44	87.82	87.98	**88.57**	85.76	85.90	79.95
MovieLens_musical	80.50	72.05	81.25	79.61	80.85	**90.34**	79.43	80.53	75.84	76.08	73.31
MovieLens_1m_age	81.96	78.79	90.34	89.80	90.43	83.16	89.92	**90.81**	87.30	87.13	83.25
MovieLens_western	84.22	89.67	84.58	86.00	85.24	**91.37**	85.72	85.82	84.94	84.84	69.12
MovieLens_horror	90.88	88.80	91.14	91.01	91.08	91.08	**91.46**	91.27	90.01	89.81	88.62
A-Card_cashout	55.90	**91.54**	74.01	70.87	70.35	83.34	71.12	70.96	90.86	90.88	90.60
MovieLens_filmnoir	79.50	71.82	76.40	78.94	76.07	**92.90**	78.57	78.60	68.33	69.10	80.17
MovieLens_war	72.86	70.61	81.18	78.82	81.71	**95.23**	79.92	80.74	79.80	79.83	77.53
LibimSeTi	99.64	99.65	99.68	**99.69**	99.68	78.97	**99.69**	**99.69**	99.65	99.65	99.66
Average Ranking	7.73	7.00	5.80	6.30	5.03	4.35	5.19	**4.10**	6.63	6.26	7.51
Number of wins	1	7	0	1	0	13	3	14	2	2	1

Table 5: Predictive performance of the models in terms of AUC for the **binary** data sets (highest-achieved performance for a data set indicated in boldface).

	MN-NB	MV-NB	LA-SVM-L2	LS-SVM-L1	LS-SVM-L2	PSN	LR-BGD-L1	LR-BGD-L2	LR-SGD-L1	LR-SGD-L2	RBF-SVM
BookCrossing	**57.24**	53.19	55.28	54.22	55.13	52.57	54.24	55.78	54.36	54.46	52.16
YahooMovies_age	64.45	65.39	64.10	64.40	64.72	65.20	64.44	**65.45**	60.94	61.19	58.83
MovieLens_crime	**74.62**	66.09	71.79	73.09	72.51	72.18	73.44	73.21	72.88	73.98	72.93
MovieLens_fantasy	71.55	51.13	63.90	62.32	62.72	**75.61**	60.09	60.35	56.68	59.53	71.81
A-Card_Goto_MAS	63.60	**76.51**	66.27	61.09	60.97	76.23	62.60	62.44	70.53	62.68	65.48
MovieLens_adventure	74.20	59.19	72.24	72.32	72.78	**76.88**	72.11	72.68	62.36	66.56	74.31
MovieLens_100k_gender	75.57	74.56	75.87	73.37	76.16	75.98	75.07	**78.72**	76.24	76.64	77.03
MovieLens_mystery	75.28	54.44	64.32	67.26	63.97	**79.02**	69.94	64.86	56.48	59.21	69.44
MovieLens_romance	71.68	55.80	67.62	65.41	67.85	**79.19**	62.69	66.23	57.86	59.28	66.14
A-Card_Goto_Permeke	77.67	80.71	65.18	68.96	67.70	79.13	69.62	67.38	81.55	79.63	**81.76**
MovieLens_drama	72.72	69.15	79.82	80.85	79.94	75.91	81.07	**81.75**	75.41	75.57	71.33
YahooMovies_gender	78.09	80.69	80.64	79.34	79.71	**82.00**	79.62	80.97	74.65	74.74	80.31
MovieLens_1m_gender	79.99	78.79	81.25	82.65	80.35	80.59	82.48	**83.23**	70.23	79.44	79.42
MovieLens_thriller	78.51	57.15	74.23	72.5	73.74	**83.97**	74.26	73.98	64.18	64.17	79.95
MovieLens_children	81.58	69.25	78.70	77.89	78.51	**83.97**	76.88	78.83	73.87	78.14	80.80
MovieLens_comedy	76.51	77.30	83.03	83.44	83.59	78.77	83.57	**84.36**	79.50	79.81	73.28
kdd2015	62.46	83.96	67.88	67.65	64.95	**84.59**	66.66	69.06	73.06	80.55	80.87
A-Card_Goto_Wezenberg	76.67	84.08	56.55	57.64	57.39	**85.07**	57.79	57.81	79.20	72.28	80.16
A-Card_defect	52.67	**85.64**	72.06	74.09	71.61	81.35	74.04	73.85	76.08	61.26	67.20
MovieLens_action	81.88	67.20	83.25	85.01	83.24	82.14	84.75	**86.12**	81.27	81.36	84.38
MovieLens_animation	84.64	67.53	**86.06**	85.86	85.62	84.17	86.13	85.45	75.16	77.10	82.97
A-Card_Goto_Roma	69.85	**86.64**	60.05	57.72	58.35	86.59	58.25	58.46	76.17	69.47	70.47
MovieLens_scifi	84.72	56.43	78.16	76.39	77.66	**87.02**	77.27	77.23	72.51	72.50	76.97
MovieLens_documentary	79.39	81.20	86.35	86.45	86.78	87.41	86.91	**87.10**	78.65	82.07	81.05
A-Card_Goto_Zoo	67.62	86.49	56.81	55.73	57.30	**87.26**	55.77	55.82	72.78	76.07	75.34
MovieLens_1m_age	80.80	78.86	85.86	**87.80**	85.18	82.95	87.79	87.71	84.53	84.84	77.57
MovieLens_100k_age	79.53	80.33	85.58	82.27	85.10	81.19	83.49	**88.52**	83.55	83.45	83.84
MovieLens_musical	88.45	57.83	80.13	79.24	80.28	**89.57**	76.83	80.22	61.82	72.64	79.84
MovieLens_horror	89.31	88.30	88.14	88.40	88.42	**90.72**	89.56	89.3	87.74	87.76	90.03
MovieLens_filmnoir	89.54	71.08	72.07	74.73	71.93	**90.78**	71.78	71.73	64.06	68.11	81.21
A-Card_cashout	53.49	91.48	71.30	66.40	66.68	86.91	66.37	66.23	86.96	80.32	**91.57**
MovieLens_western	89.04	66.81	82.92	83.61	83.71	**91.68**	83.64	84.00	74.31	81.05	87.46
MovieLens_war	83.20	65.55	75.83	75.82	78.06	**94.06**	82.07	73.68	65.28	67.73	73.97
LibMiSeTi	99.64	99.65	99.68	**99.69**	99.68	78.97	**99.69**	99.68	99.65	99.65	98.78
Average Ranking	5.73	7.70	5.97	6.39	5.79	**3.58**	5.66	4.76	7.72	7.36	5.55
Number of wins	2	3	1	2	0	16	1	8	0	0	2

Table 6: Predictive performance of the models in terms of AUC for the **numeric** data sets (highest-achieved performance for a data set indicated in boldface).

4.1.2 Comparison of Binary and Numeric Behavioral Data

Binary and numeric data are contrasted with the Wilcoxon signed-rank test to determine if numeric behavioral information, which models strength of behavior, leads to better predictions. This leads to the finding that LS-SVM-L2, LA-SVM-L2, LS-SVM-L1, LR-SGD-L1 and LR-SGD-L2 in fact perform better for binary data sets. A tendency towards a similar result was found for LR-BGD, but without statistical support. Discriminative, linear classifiers which try to maximize the distance between a hyperplane and support vectors are very sensitive to the distance of the instances along the feature axes (Forman et al., 2009). It is therefore beneficial for those models that the range of distances along each feature axis is small and this is the case when only taking into account binary information (Hsu et al., 2003). Thus, presence/absence information is sufficient evidence of instances' class membership for these discriminative techniques.

Additional significant differences contrasting binary and numeric data on computational performance are the following. All support vector machines (LA-SVM-L2, LS-SVM-L1 and LS-SVM-L2) and both batch gradient descent logistic regression methods (LR-BGD-L1 and LR-BGD-L2) run faster when faced with binary behavioral data. In contrast, MN-NB runs faster when faced with numeric behavioral data.

4.1.3 Interpretation of Performance Analysis

First, let's focus on Naive Bayes. Overall, the results corroborate what has been found in prior predictive analysis studies: given enough data, discriminative classifiers outperform generative classifiers (Ng and Jordan, 2002). Because of the constraints of the event generation schemes, naive Bayes is not able to grasp the underlying distribution of the data as well as (say) logistic regression. This suggests the need for context-specific event models for behavioral data, as neither the multivariate Bernoulli nor the multinomial event model is quite right for many behavioral data settings (Junque de Fortuny et al., 2013).

Separately, the naive Bayes results also raise a warning for researchers and practitioners. The importance of considering the assumptions of the underlying data-generating process is underlined when comparing MV-NB and MN-NB. It is striking that MN-NB performs quite poorly (see Table 6), but at the same time belongs to the top-used techniques in relevant literature, as illustrated in Table 1. We conjecture that the likely reason for its frequent use is that multinomial NB has been shown to perform well in settings such as text mining (Hand and Yu, 2001). Since these behavioral data resemble (perhaps superficially) text data sets, one might conclude that naive Bayes would also work well here. For binary behavioral data, however, MV-NB performs better than MN-NB. In text classification, comparisons between both event models on binary bag-of-words data have shown superior performance of the multinomial event model (although no evidence for statistical significance was found) (Schneider 2004; Metsis et al. 2006). The reason for this can be attributed to the assumptions made by the underlying event models. The multivariate event model assumes each feature to be generated by independent boolean draws and thus models the presence and absence of features (McCallum and Nigam, 1998). In contrast, the multinomial model captures frequencies of features and assumes the features to be drawn independently and with replacement from the collection of all features. When modeling binary data, the former appears the best fit. The fact that this does not hold for text data reinforces the importance of context-specific modeling schemes and stresses that care should be taken in assessing which event generation model best fits the analyzed data (Junque de Fortuny et al., 2013). This is also in line with the no-free-lunch theorem stating that no assumptions can be made regarding a classifier's superior performance across different data contexts (Wolpert, 1996). Currently, for human behavioral data our result suggests that the multinomial event model is not well suited (Junqué de Fortuny et al., 2013; Junque de Fortuny et al., 2013).

In contrast, for numeric behavioral data where the features model the strength of an action, the multinomial model outperforms the multivariate event model. This follows the intuition behind their underlying event models as stated above. In text classification, MV-NB was not found to perform well on numeric data. Also, by running MV-NB on an expan-

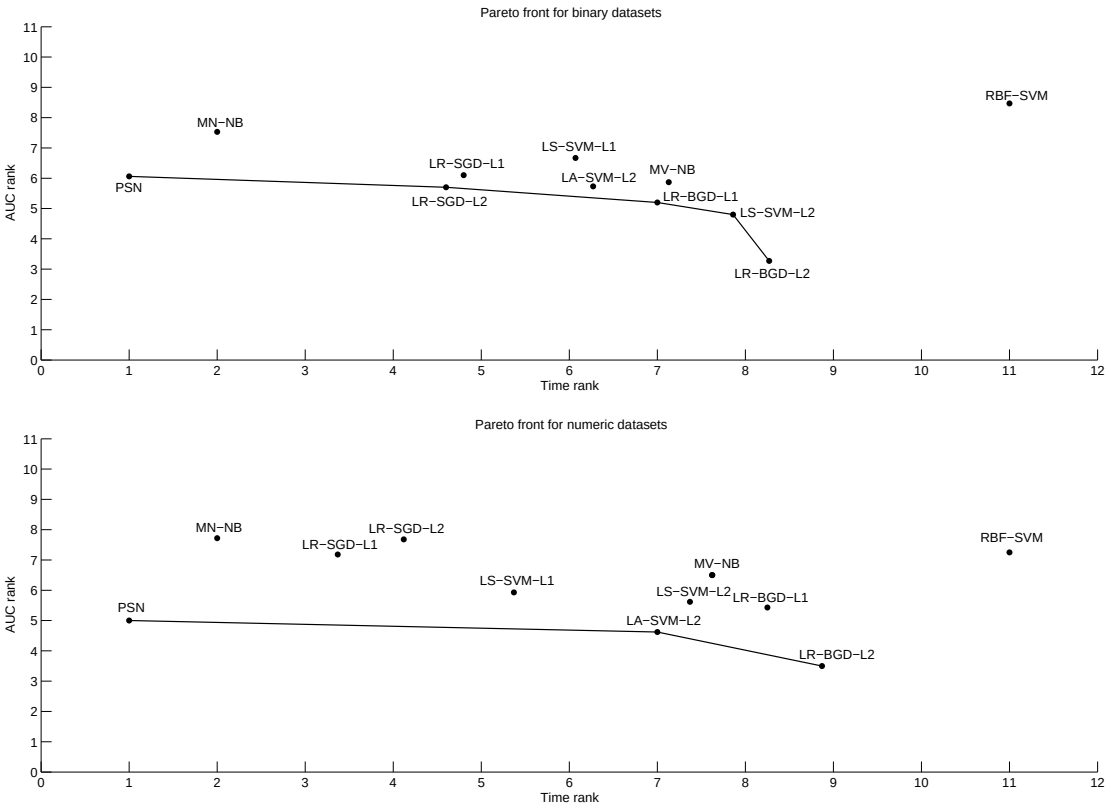

Figure 7: Pareto front for binary (top) and numeric (bottom) behavioral data.

ded unary-encoded data set, increased dependence between features is created, violating the underlying model assumptions and resulting in lower performance (Bermejo et al., 2014).

In text classification studies, MN-NB is seldom used with feature values that model absolute frequencies of words (as they do here). Typically, an inverse document frequency (IDF) measure is employed, favoring less frequently occurring words. This IDF philosophy is incorporated in the PSN method and weighting features in this manner thus seems beneficial in a behavioral context. Approaching the problem through a weighted bigraph between persons and behaviors is a very intuitive approach on the one hand, but also results in the most time-efficient method.

Moving on to the discriminative classifiers, we observe the superior performance of L2 regularization over sparse L1 regularization. This confirms the findings of Zhu et al. (2003) and Bannur (2011). Moreover, Ng (2004) theoretically shows that due to the rotationally-invariant nature of L2 regularization, it is better suited than its L1 counterpart in a context with many relevant features. This implies that many features in the high-dimensional behavioral context contribute to the prediction, which is also confirmed in the learning curve analysis (below). Taking into account people's limited behavioral capital and the very sparse feature vectors, it makes sense to learn dense concepts (from the sparse data). Similar results have been found elsewhere for behavioral data (Clark and Provost, 2016), and also for the analysis of high-dimensional text data (Joachims, 1998).

Comparing the performance of linear versus non-linear classifiers, it can be observed that the linear classifiers overall perform better. This is in line with results from text classification (Joachims, 1998), where the problem is often considered linearly separable due to the high-dimensionality. In that case, unnecessarily transforming the problem to a higher-dimensional feature space with RBF-SVM leads to overfitting. For numeric data, however, RBF-SVM performs better compared to its performance on binary data which can be attributed to the fact that the discriminative linear classifiers do not perform

well with numeric data as stated above. With a larger range of feature values, the space in which a hyperplane is to be found becomes much larger, increasing the effect of the curse of dimensionality and decreasing the possibility of linear separability. The burden of going through much more computational complexity, however, generally does not justify its use in such high-dimensional settings, as the results here corroborate.

Further, the Pareto fronts confirm our previous findings. Clearly, classification techniques such as NB, PSN and SGD, which all make use of strong assumptions to simplify the classification process are located most left in Figure 7. Making these assumptions (for example that the features are conditionally independent in the case of NB) leads to lower runtime complexity, while at the same time giving up AUC. Only MV-NB presents an exception to this: its runtime of $\mathcal{O}(\bar{m} \cdot n)$ approaches that of the SVM techniques. On the other hand, MV-NB also consistently performs best on large, imbalanced data sets. Although the specific generation process of human behavioral data is not known, estimating simple models or heuristics (in the spirit of backward-engineering the data generation process) can provide insight into this process as well as provide decent (and even very good) results when modeling this data (Gigerenzer and Goldstein 1996; Gigerenzer et al. 1999; Green and Armstrong 2015).

PSN offers an interesting case when we look across all of the results. As just noted, it is by far the fastest technique. In terms of predictive performance, although in the main comparative analysis its rank is middling, note that it is not statistically significantly worse than the best performer. Also note that when we go back to the full studies, PSN dominates in wins for the numeric data sets and is a very close second for the binary data sets. This difference from the main results is because PSN consistently reaches the best AUC across the *MovieLens_genre* data sets, most of which are essentially discarded for the main analysis.

4.1.4 Summary of Performance Analysis

We summarize the conclusions of the performance analysis as follows:

- Overall, discriminative classifiers perform better than generative classifiers with LR-BGD-L2 yielding the best generalization performance (AUC) for both binary and numeric data. However, in terms of computational efficiency, it performs worst.
- In general, L2 regularization performs better than L1 regularization. Table 1 demonstrates that a presumption towards this empirical finding exists in literature as L2 regularization is used more frequently. As a drawback, we find that L2 regularization takes more time.
- BGD optimization is slower than its SGD variant, while resulting in better generalization performance.
- RBF-SVM is the slowest method for both binary and numeric data. This is also stated in many papers as the reason why RBF-SVM is not considered an option with such high-dimensionality.
- PSN and MN-NB build their models in the shortest amount of time with the heuristic assumptions of PSN leading to better results in the smallest amount of time.
- MV-NB performs better than MN-NB for binary data; the opposite holds for numeric data.
- If you want a fast and moderately accurate method, PSN is the way to go. It is by far the fastest, and its predictive performance is fairly good.
- Contrasting binary and numeric behavioral data, LIN-SVM and LR result in better predictions in a lower amount of time for binary data. On the other hand, MN-NB achieves better run time for numeric data.

Linking these results to Contribution I of this work, we can thus conclude the following. LR-BGD-L2 performs best in terms of AUC on binary behavioral data sets. However, in a practical setting, its time complexity might render it impracticable. An attractive trade-off between performance and time is given by the PSN technique. Regarding Contribution II, for the discriminative linear SVM and logistic regression classifiers, the mere modeling of presence and absence of features (binary features) is superior both with respect to classification performance as well as computational runtime.

4.2 Learning Curve Analysis

Figure 8, Figure 9 and Figure 10 show the learning curves in the instance dimension, in

the feature dimension for random feature selection, and in the feature dimension with feature selection for the largest behavioral data sets. In order to provide clarity to the many learning curves, we structure the analysis to find general patterns and attempt to identify groups of similar behavior. Note that some learning curves demonstrate deviant behavior from the general trend, which is mostly due to the random sampling procedure leading to varying imbalance and sparsity levels. In Appendix B, more learning curves for these dimensions can be found. Not all learning curves are shown; however, we have selected representative learning curves showing the main behaviors present in the data collection.

4.2.1 Instance Dimension Learning Curves

For the majority of the learning curves, it can be seen that the SVM classifiers overall show similar behavior, with L2 regularization often dominating L1 regularization. Moreover, the LR-SGD curves are often similar to one another. These findings confirm our previous results.

On a more detailed level, we attempt to relate the shape of the curves to data characteristics. Four cases can be identified dependent on two dimensions: signal-from-noise separability and imbalance. The exact signal-from-noise separability of a data set cannot be determined, so a proxy $\widetilde{SNS}$ is used in the form of the maximum AUC reached by the classification techniques analyzed here, analogous to Perlich et al. (2003). Two cases are distinguished: $\widetilde{SNS} \leq 83\%$ refers to lower signal separability while $\widetilde{SNS} > 83\%$ refers to higher signal separability, which is essentially the same split used by Perlich et al. (2003). The second dimension denotes the imbalance of the target variable. A high imbalance is recorded if less than 5% of the labels are positive, otherwise imbalance is considered low.

Along these two dimensions, four cases can now be discussed. The *first* case is characterized by low imbalance and high separability. The most obvious illustration can be seen in *LibimSeTi*. An instance sample of less than 1% is sufficient to reach an AUC not significantly different from the final performance. The techniques learn fast and the result is a concave-down learning curve. MV-NB is the only technique which requires considerably more instances to learn from this data as demonstrated by the later occurrence of this shape. As the signal-from-noise separability decreases, but stays above 83%, the curves stay concave down (*MovieLens_action*, *MovieLens_comedy*, *MovieLens_horror*. This is the case for data sets where feature dimensionality is lower and the curves thus resemble traditional concave-down learning curves and learning curves for textual data (Colas and Brazdil, 2006). The *second* case is illustrated by *MovieLens_scifi*, *MovieLens_thriller* and *MovieLens_western*. Here, the separability is still high but the imbalance also is high resulting in concave-up learning curves.

Thirdly, when the separability is low and imbalance is high, a concave-up curve is observed and generative techniques demonstrate more robustness towards that imbalance (*Fraud*, *Car* and *Banking*). Theoretically indeed, SVMs are not able to generalize well with high imbalances as a separator is learned which is biased towards the minority class (Liu et al. 2007; Wallace et al. 2011). Moreover, combining small instance samples with low evidence of positive samples, discriminative models have more difficulty to separate the instances (Ng and Jordan, 2002). *Lastly*, when both separability and imbalance are low (*TaFeng*, *BookCrossing* and *YahooMovies*), again a concave-up/linear curve demonstrates a slow start-up in learning. Also, all classifiers demonstrate comparable behavior in attempting to capture the low-separable signal in the data.

In summary, in all cases except the first (low imbalance, high separability), the learning curves show concave-up/linear behavior, which implies that for these behavioral data sets, for the ranges that we are able to consider (which in some cases are quite large), adding more training instances keeps on yielding substantive increases to the classification performance. Although of course there is an inherent ceiling on predictive performance, in many of these cases there still seems to be room for significant improvement. This is in contrast to learning curves for large traditional, non-behavioral data (such as Shavlik et al. 1991; Perlich et al. 2003; Martens et al. 2016) which mostly have concave-down shapes. In that case generally, the benefit of adding training set samples leads to diminishing return in AUC (Provost and Kolluri, 1999). Importantly, the com-

monality of concave-up learning curves should lead practitioners to exercise caution when performing pilot studies on smaller data samples—as the observed performance may well not represent what is possible to achieve larger data sets.

Linking this to Contribution III of this work, it is apparent from these learning curves that overall adding more training instances leads to a better performing model. This reinforces what has been found in Junqué de Fortuny et al. (2013): even for large data sets, more data indeed still often will yield substantially better predictions.

4.2.2 Feature Dimension Learning Curves with Random Feature Selection

It can be seen that all algorithms start roughly equally when faced with few features and overall no classification technique is significantly better at handling fewer features.

When the data has low separability and no extreme imbalance (*YahooMovies*, *TaFeng* and *BookCrossing*), the learning curves are concave up and no classifier dominance can be distinguished. As imbalance increases for low $\widetilde{SNS}$ datasets (*Banking*, *Car* and *Fraud*), the generative classifiers again dominate the discriminative techniques. The latter also holds for highly separable datasets (*MovieLens_scifi*, *MovieLens_thriller* and *Acard_Wezenberg*), although the high $\widetilde{SNS}$ and the lower feature dimension reshape the end of the learning curves towards being concave down. In the case of a high signal from noise separability with no extreme imbalance on large datasets (*Flickr* and *KDDa*), the techniques learn slowly resulting in concave-up curves. In contrast, for smaller datasets (*MovieLens1m*, *KDD2015* and *MovieLens_horror*) learning goes faster, and the resulting curves are linear or concave down.

A comparison of the learning curves in the instance and feature dimensions leads to the finding that performance convergence is more sensitive to the features than to the instances: the feature learning curves overall demonstrate concave up behavior. This is strongly confirmed in the high $\widetilde{SNS}$ data set *LibimSeTi*: the performance converges faster in the instance dimension than in the feature dimension.

Following the findings in the previous section, the support vector machines and the SGD variants each demonstrate similar behavior. Regarding individual classifiers' robustness, it is hinted at by the learning curves for *MovieLens (comedy)*, *MovieLens1m (age)* and *LibimSeTi* that MV-NB learns more quickly in the feature than in the instance dimension: the bias component of its error is larger in the latter due to smaller sampling size combined with having many features (Friedman, 1997).

The foremost conclusion from these results is that adding more features leads to higher predictive performance (Contribution III). Moreover, due to the shapes being concave up (and some linear), it seems that many features provide significant, independent predictive evidence. We look deeper into this next.

4.2.3 Feature Dimension Learning Curves with Intelligent Feature Selection

Unsurprisingly, it can be observed that the starting point with intelligent feature selection is higher in comparison to the starting point when adding random features. For the large behavioral data sets (very fine-grained with more than 1 million features such as *Car*, *KDDa* and *Banking*), adding more features, the curves exhibit a similar concave-up shape as when no feature selection is used. This prompts us to conclude that for these very large behavioral data sets, the features show low redundancy and very many are essential in predicting the target variable. For smaller behavioral data sets (such as *Flickr*, *BookCrossing*, *TaFeng* and *Ecommerce*), the curves change from concave up to linear. Hence, there is discriminative informative value present in the features, although each still contributes to better predictions. This has also been found in text analysis (Joachims, 1998). For the other data sets, the curves demonstrate concave-down learning behavior. Adding the most informative features first leads to a significant performance increase. The remaining less-discriminative features result in diminished increases and in some cases even decrease the AUC. This is the case for the linear support vector machines in a high-imbalanced setting (*Acard_Permeke*, *Acard_Wezenberg*) for which these techniques are highly sensitive (Forman, 2003). Similar results have been found in text classification where the feature dimensions are comparable in size

to these lower feature-granularity data sets (Colas and Brazdil, 2006).

Referring back to Contribution III, using more, although less informative features still leads to a higher predictive value, especially in the case of very fine-grained features. This implies that care should be taken with preprocessing techniques such as feature selection in the context of very fine-grained behavioral data.

5 Conclusion

The academic literature regarding big behavioral data provides substantial evidence of its predictive power in a wide variety of fields. However, not all state-of-the-art classification techniques are suitable for the high-dimensional and sparse characteristics of these data sets. Through a systematic comparative benchmarking study, this paper investigated the performance of these state-of-the-art classifiers with large, sparse behavioral data.

The first contribution consists of finding a well-performing method both in terms of AUC and computational complexity. The results, however, indicate that an AUC-time trade-off is inherent to the problem as the Pareto front clearly illustrates: given more time, one can choose to achieve higher classification performance. In terms of AUC, logistic regression with L2 regularization leads to significantly better results. Unfortunately, it attains this result at a high computational cost. Relating these results to the techniques used in the academic literature (Table 1), linear support vector machines are most frequently used, while our results find that logistic regression would perform better. The propensity in literature towards the use of L2 regularization is supported by our findings.[20] This suggests and confirms findings from comparisons with dimensionality reduction techniques that each behavioral feature captures a different fine-grained aspect of an instances' behavior, resulting in low feature redundancy. The learning curves built by adding features dependent on their information value also support this finding. In terms of computational complexity, PSN and MN-NB stand out with their significantly low run time. MN-NB is commonly used (see Table 1) due to its frequent and successful application in high-dimensional text analysis. Despite its speed, however, its underlying assumptions do not lead to high-quality predictions in this behavioral setting. PSN appears to be a much better AUC-time trade-off.

On a more fine-grained level, a tree classifier and a logistic regression are learned on the results to explore the competence domain of the best-performing classifiers. These meta-analyses are to be interpreted with caution due to the restricted sample size. As imbalance increases, MV-NB performs better. If the sample is heavily unbalanced for small data sets, PSN becomes the method of choice. As confirmed in the learning curves, the generative techniques indeed perform better in a highly imbalanced setting. In low-imbalanced data sets, the discriminative classifiers have higher AUC.

The second contribution is to determine whether a more complex numeric representation of the behaviors adds predictive power over a binary action-taken-or-not representation. The discriminative techniques perform better when the data merely models presence/-absence of features in contrast to data enriched with behavioral strength; the mere presence of behavior apparently informs the modeling sufficiently, which obviously could lead to decreased investment in data collection and management.

By systematically comparing these classification techniques in a benchmarking study, we have formally investigated what is correctly or incorrectly presumed by previous behavioral analysis studies. The conclusions can now point researchers and practitioners towards a unifying direction for both future behavioral research and future technique optimization research. Furthermore, the importance of context-specific data modeling schemes has been emphasized.

Limitations related to this analysis originate from limited public availability of behavioral data sets. This results in a relatively small sample for significance testing. However, we worked to make it as broad a sample as possible. One avenue of future research therefore consists of updating this proposed set of benchmarks with even more behavioral data

[20] This is in contrast to what, at least anecdotally, is a common reaction among practitioners: that since the data are high-dimensional and sparse, one should use L1 regularization, confusing the presence of sparse data with a desire for a sparse model.

sets as these become available. Especially for numeric data sets, this could lead to stronger conclusions. Moreover, sound meta-analyses could then strengthen the relations between data set characteristics and choice of classifier. Ideally, in future research an event generation model can be constructed for this type of data resulting in the generation of artificial data sets that could enrich this benchmark collection. A second possibility for further research constitutes a focus towards scaling up the well-performing L2-regularized techniques in terms of computational complexity on very sparse data (Dalessandro et al., 2014), most importantly for LR-BGD-L2. It would also be interesting to explore whether fast, heuristic predictions by e.g. a PSN technique could be used to speed up the training phase of more complex, slower classifiers such as LR-BGD-L2 or RBF-SVM in order to combine the best of both worlds (Dalessandro et al. 2014; Junqué de Fortuny et al. 2015).

With respect to the third contribution, a learning curve analysis is performed which shows that better performance continues to be observed when more data (both in instance and feature dimension) is used. In contrast to non-behavioral instance learning curves, the curves are generally linear/concave-up. This implies that performance still increases when adding more training data, even to very large data set sizes, which is only marginally the case for traditional non-behavioral features. Very fine-grained, large data sets which demonstrate very low redundancy in the features show no dependence on the informative value of features as demonstrated by the concave-up curves when adding informative features in a descending fashion. For smaller, less fine-grained data sets, a higher redundancy between the features is present with a higher sensitivity to more behavioral features. Hence, it is very valuable to collect as much data as possible in this behavioral setting both regarding more instances as well as regarding more modular behavioral aspects.

Moreover, this shows that traditional learning curve analysis might be misleading. For example, Provost and Fawcett (2013) suggest that investing in more training data probably is not worthwhile as learning curves show that generalization performance has leveled off. This advice was indeed supported by traditional learning curve analyses, where one seldom witnesses learning curves that look poor for significant stretches and then suddenly turn steeply up. However, here we see this pattern repeatedly and thus researchers and practitioners should be given different advice for data such as these. For future research, defining and quantifying behavioral characteristics of data sets could also prove helpful in determining causes for different generalization patterns of classification techniques.

In this paper we focused on the predictive performance (and computational requirements) of the generated classification models. Increasingly important aspects of such models is explainability and fairness. The ability to explain the decisions made can be important for various reason such as model acceptance and model improvement, see for example Martens and Provost (2014). The high-dimensionality of behavioral datasets make traditional approaches, such as investigating the coefficients of a linear model or rule extraction, problematic. Instance-based approaches might be an interesting alternative for this setting, to be investigated in future research. Related to this issue, detecting and removing potential negative bias against sensitive groups (defined by for example gender, race or sexuality) to ensure fairness of the prediction model also constitutes a relevant and challenging issue for future research.

As a final conclusion, it is apparent that the predictive analysis of big behavioral data significantly differs from the analysis of traditional (even big) data. The results of this study should be taken into account in the general predictive analysis of this kind of data.

A Classification Techniques

A classification technique takes a data set X along with values Y for the target variable for each of the instances $\mathbf{x}_i$ in X and attempts to learn a function $h(\mathbf{x}) = \hat{y}$ as an approximation of the true value Y. The classifier builds a predictive model based on a training set (X_{train}, Y_{train}). The trained model is then used to predict y values of new, unseen data points belonging to a test set.

A.1 Naive Bayes

The naive Bayes classifier is a generative classifier using Bayes' rule to build a predictive model

$$p(y|\mathbf{x}) = \frac{p(y)p(\mathbf{x}|y)}{p(\mathbf{x})}.$$

Since the denominator is not dependent on the class variable y, it is not taken into account. Then, making use of the naive assumption that features are mutually conditionally independent, the above equation can be rewritten as follows and forms the probability model used by the naive Bayes classifier

$$p(y|\mathbf{x}) \propto p(y) \prod_{j=1}^{m} p(x_j|y). \quad (1)$$

In order to determine $p(\mathbf{x}|y)$, an underlying event model is assumed for the generation of the features. Considering the binomial and multinomial character of the distributions of the behavioral features, the multivariate and the multinomial event model are considered suitable.

A multinomial event model has proven successful in text classification, an area also characterized by high dimensionality (McCallum and Nigam, 1998), and this model defines the conditional probability as

$$p(\mathbf{x}|y) = \frac{(\sum_{j=1}^{m} x_j)!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} p(x_j|y)^{x_j}.$$

A multinomial distribution implies that the features result from independent draws from the collection of all features. It does not take into account absent features, which is computationally beneficial in a sparse context. The training time complexity of its implementation consists of calculating a vector of feature weights for each class and results in $\mathcal{O}(m)$ time.

The multivariate event model defines the conditional probability as

$$p(\mathbf{x}|y) = \prod_{j=1}^{m} p(x_j|y)^{x_j}(1 - p(x_j|y))^{(1-x_j)}.$$

Theoretically, this event model excellently lends itself to binary data: a feature is present with probability $p(x_j|y)$ and absent with probability $1-p(x_j|y)$. However, since the absence of features is explicitly modeled, its implementation is not naturally tailored to sparse data. Therefore, an efficient sparse implementation presented by Junqué de Fortuny et al. (2013) is used. This implementation takes advantage of the assumption that the features are binary and transforms Equation 1 into

$$\log p(y|\mathbf{x}) \propto \log p(y) + \sum_{j|x_j=1} \log p(x_j = 1|y) + \sum_{j|x_j=0} \log(p(y) - p(x_j = 1|y)).$$

This transformation results in a $\mathcal{O}(\bar{m} \cdot n)$ time complexity in contrast to $\mathcal{O}(m \cdot n)$ with $\bar{m}$ the number of active elements.

A.2 Logistic Regression

In logistic regression, the target function, $h(\mathbf{x}) = (\mathbf{w}^T\mathbf{x})$, is transformed with the use of the logistic function with $\mathbf{w}$ a vector of weights corresponding to the dimensions of X. This transformation models a probabilistic estimate as to whether a test instance belongs to the positive class. The logistic regression model is thus defined as

$$p(y|x) = \frac{1}{1 + e^{-yw^Tx}}.$$

When training the logistic regression model, the function

$$\min_w R + C \sum_{i=1}^{n} log(1 + e^{-y_i w^T x_i}), \quad (2)$$

is optimized, where R is the regularization term to prevent overfitting. With L1 regularization, the value of R equals $||\mathbf{w}||_1$, with L2 regularization, R is $\frac{1}{2}||\mathbf{w}||_2^2$. The former regularization parameter zeroes out low-valued coefficients which results in natural feature selection (Ng, 2004). The latter, in contrast, favors very small, non-zero weight values. This regularization is controlled by a parameter C which models a trade-off between the complexity of the model (first term) and minimization of the training error (second term). Extremely minimizing the training error might result in a complex model with lower generalizability which the regularization parameter C attempts to correct.

In the search for an optimal $\mathbf{w}$, Equation 2 can be solved with Newton's methods batch gradient descent (LR-BGD variants) or with stochastic gradient descent (LR-SGD variants) (Bottou, 2010). The LIBLINEAR package implements logistic regression with a trusted region Newton method (Fan et al. 2008; Lin et al. 2008). Iteratively, a subset of the region of the objective function is approximated and subsequently expanded or shrinked depending on the quality of the approximation. This is done in $\mathcal{O}(\bar{m} \cdot c)$ time where c is the number of iterations needed until convergence. Stochastic gradient descent is scalable towards larger dimensions since it approximates the true gradient of $\mathbf{w}$ by calculating the gradient over one random training instance. This approximation is seen as a proxy for the real gradient and is used in subsequent steps of the algorithm. While the execution time decreases, clearly, convergence towards an optimum value will be slower. VOWPAL WABBIT, a widely used analysis tool for big data, solves the LR-SGD variants with stochastic gradient descent in $\mathcal{O}(n)$ time (Langford et al., 2007).

A.3 Support Vector Machine

The support vector machine (SVM) (Cortes and Vapnik, 1995) is a discriminative binary classifier that is very suitable for high-dimensional data. An SVM finds a hyperplane that maximally separates the closest points of each of two classes, called support vectors (SV). In maximally separating the SVs, the SVM aims for high generalizability and low

variance. If the data is separable, the hard margin SVM seeks a hyperplane of the form

$$\mathbf{w}^T\mathbf{x} + b = 0,$$

with $\mathbf{w}$ the weight vector normal to the hyperplane and b a bias. New test points are classified on one of the sides of this hyperplane, i.e.

$$\begin{cases} \mathbf{w}^T\mathbf{x}_i + b \geq +1 & \text{if } y_i = +1, \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 & \text{if } y_i = -1. \end{cases}$$

When faced with non-linearly separable data, a non-linear function $\theta(\mathbf{x})$, called a kernel, is used to project the data points to a high-dimensional feature space where the points are linearly separable. In general, the goal of the support vector machine is to solve the objective function

$$\begin{aligned} \min_{w} b, \xi \quad & R + C\sum_{i=1}^{n} \xi_i, \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\theta(\mathbf{x}_i + b)) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \end{aligned}$$

with C once again a trade-off parameter between complexity (first term) and error rate (second term) and ξ_i $(i = 1, ..., n)$ slack variables representing the loss function. In words, the goal is to minimize the training error (second term), while allowing for misclassifications (first term), regulated by the trade-off parameter C. Three parameters are to be defined in the above equation, i.e. the regularization parameter R, the loss function ξ_i and the kernel function θ.

The first parameter can be defined following L1 or L2 regularization. In the first case, $R = ||\mathbf{w}||_1$ and in the second case, R is equal to $\frac{1}{2}||\mathbf{w}||_2^2$. As mentioned before, L1 regularization results in sparse outputs. For the loss function ξ, the L1-norm and the L2-norm are considered here. Selecting the L1-norm as loss function ξ, the sum of the absolute differences is minimized: $\xi_i = \max(0, 1 - y_i\mathbf{w}^T\mathbf{x}_i)$. When using the L2-norm as loss function ξ, the square of the errors is minimized and can be defined as follows: $\xi_i = \max(0, 1 - y_i\mathbf{w}^T\mathbf{x}_i)^2$. Since the second loss function attempts to minimize the squared errors, it is more sensitive to outliers. Regarding the kernel, two options are explored: a linear kernel and an RBF kernel. A linear SVM uses a linear function as the kernel $\theta(\mathbf{x})$. It is often stated in literature that with high-dimensional data a projection to a higher-dimensional feature space to find a hyperplane will come at too high a computational cost and will not improve classification performance (Hsu et al. 2003; Yu et al. 2010). RBF-SVM, on the other hand, uses a non-linear kernel and is capable of capturing complex interactions in the data (Chang and Lin, 2011). An RBF-SVM operates with a Gaussian kernel and takes the form

$$K(\mathbf{x}_i, \mathbf{x}'_j) = e^{-\gamma||\mathbf{x}_i - \mathbf{x}'_j||^2},$$

with $\mathbf{x}_i$ and $\mathbf{x}'_j$ two samples of which the Gaussian kernel determines the similarity in the new high-dimensional space guided by parameter γ. The parameter γ controls the standard deviation of the Gaussian at each point: the higher a value for γ, the lower the influence of the SVs which decreases bias, but increases variance. The LIBLINEAR package (Fan et al., 2008) is used for the implementations of the different variants of linear SVM (i.e. LS-SVM-L2, LS-SVM-L1, LA-SVM-L2). It uses a coordinate gradient descent method solving the optimization problem in $\mathcal{O}(n)$ time. For the RBF-SVM, the LIBSVM package (Chang and Lin, 2011) is used which leads to a training time complexity that scales between $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$.

A.4 Relational Classification with Pseudo Social Networks

In this approach, the data is transformed to a similarity network (pseudo social network, PSN) between the instances (Stankova et al. 2014; Martens et al. 2016). The network is denoted 'pseudo' as no true social network is implied: two instances are connected if they are similar regarding behaviors they have engaged in. Based on this similarity, predictions are made using traditional relational classifiers. Concretely, first, weights are calculated with a top-node function for each feature based on its degree (Stankova et al., 2014). We employ the tangens hyperbolicum which defines the weight s_m for a feature m as

$$s_m = tanh(\frac{1}{d_m}),$$

with d_m the degree of node m such that features with a low degree receive a higher weight. Then, the pseudo social network is built by connecting instances, weighing their edges based on their shared features. The feature weights are aggregated into edge weights w_{ij} between nodes i and j through an instance node function. The sum of shared nodes function simply sums the feature weights s_m of the shared features of instances i and j as

$$w_{ij} = \sum_{m \in N(i) \cap N(j)} s_m,$$

with $N(i)$ the features demonstrated by instance i. Now, relational classifiers are used. These classifiers infer unknown labels through network structure and labels of connected nodes. We use the weighted-vote relational neighbour classifier (Macskassy and Provost, 2007), which labels a node through a weighted probability estimation using the known labels of connected nodes. Formally, the classifier calculates

$$P(l_i = c|N(i)) = \frac{1}{Z}\sum_{j \in N(i)} w_{ij}P(l_j = c|N(j)), \quad (3)$$

with l_i the label of node i, $N(i)$ the instance nodes connected to node i and Z the number of connected nodes. In Stankova et al. (2014), a highly-scalable version of the combination of the sum of shared nodes instance node function with the weighted-vote relational classifier is deduced resulting in a fast linear model over the feature nodes, referred

to as SW-transformation. This fast, scalable variant with $\mathcal{O}(m)$ runtime complexity lends itself excellently in the context of sparse, high-dimensional data and translates Equation 3 into

$$P(l_i = c|N(i)) = \frac{1}{Z} \sum_{m|x_{im} \neq 0} ns_m \times s_m,$$

where $ns_m = |x_{jm} = 1 \text{ and } y_j = 1|$ and s_m is the weight of top node m and $N(i)$ the instance nodes connected to node i. For a full account of this method, we refer to Stankova et al. (2014).

A Python implementation of the SW-transformation is available[21].

B Learning Curves

[21] `https://github.com/SPraet/SW-transformation/`

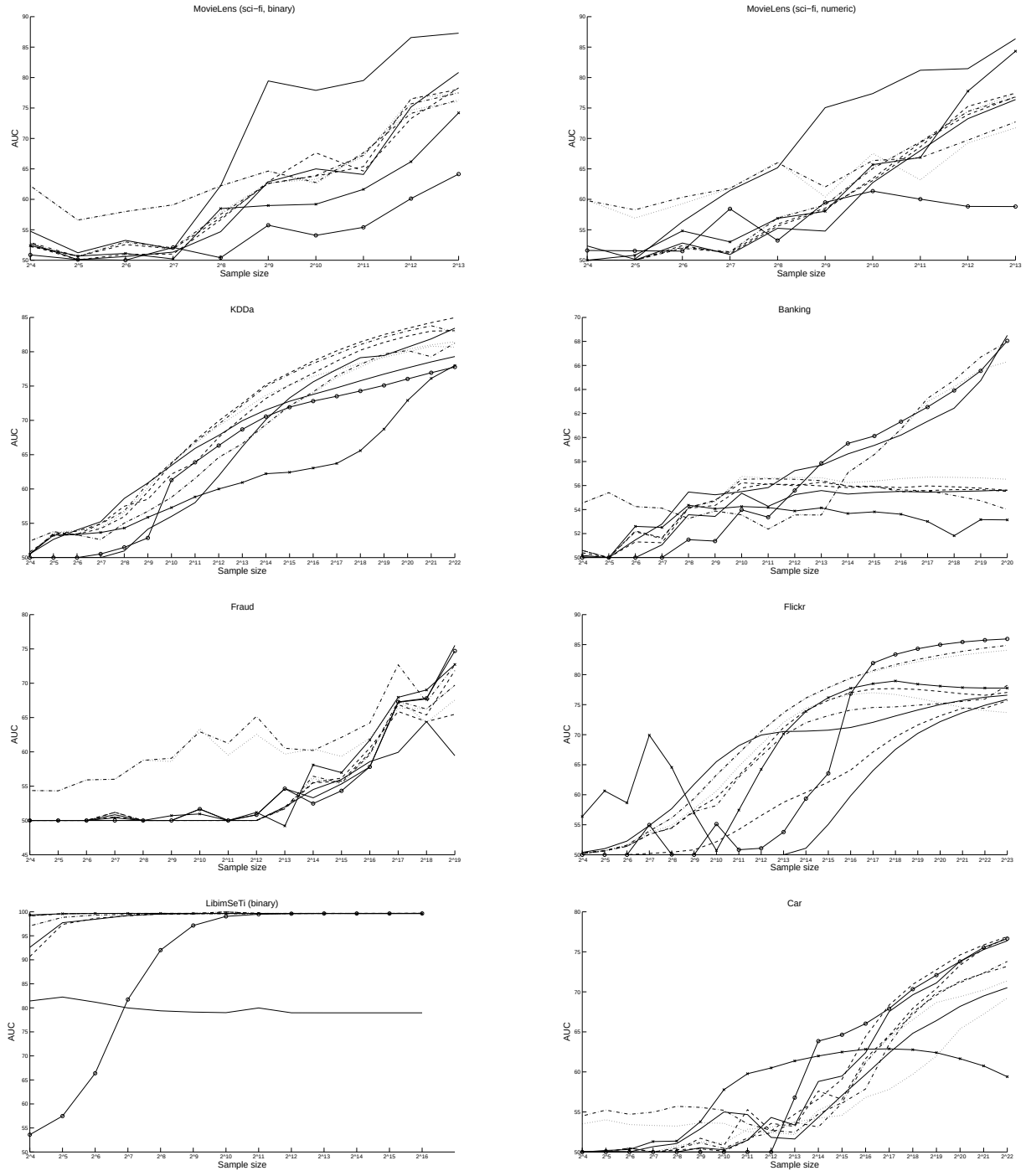

Figure 8: Learning curves in the instance dimension.

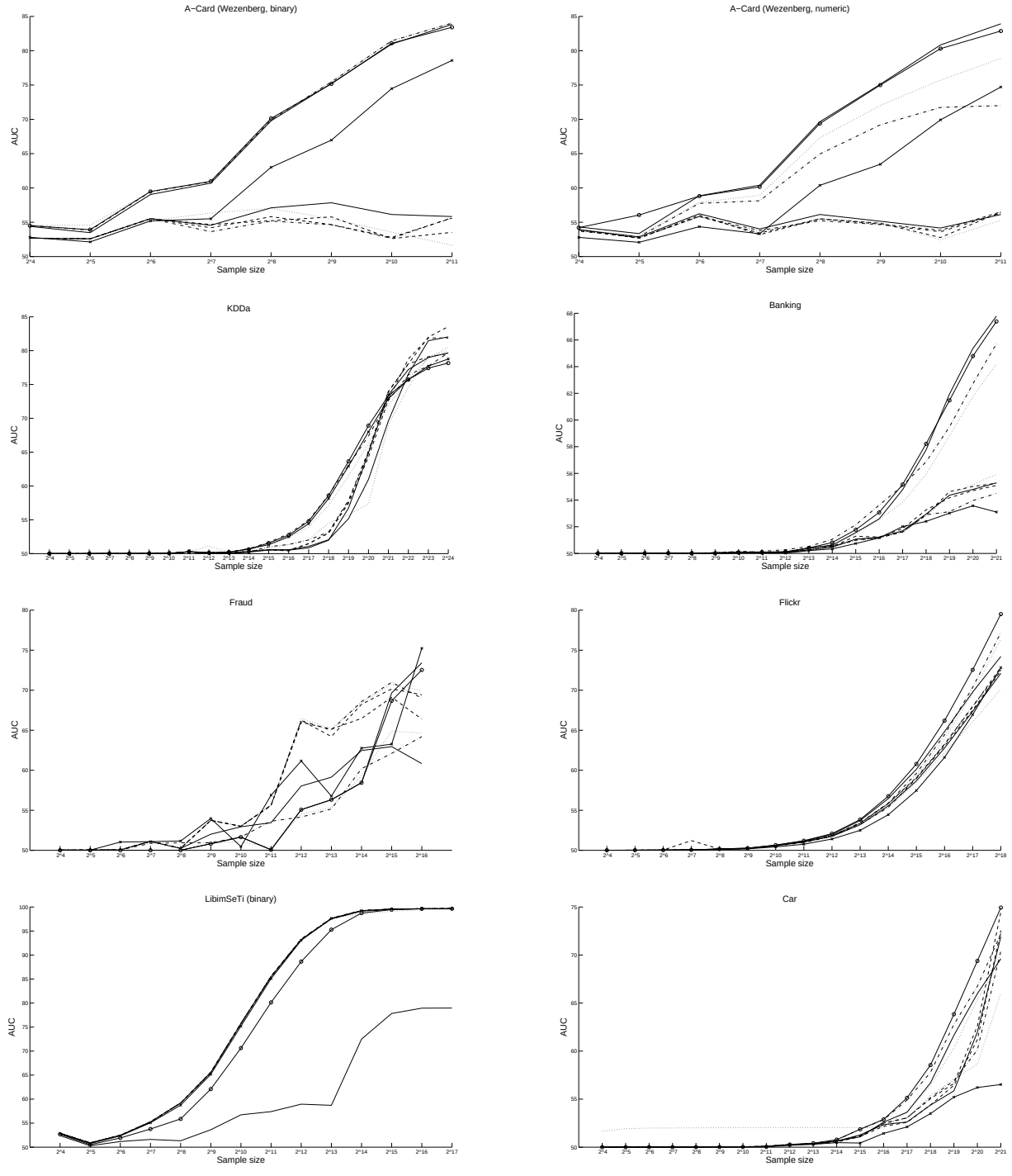

Figure 9: Learning curves in the feature dimension (random feature selection).

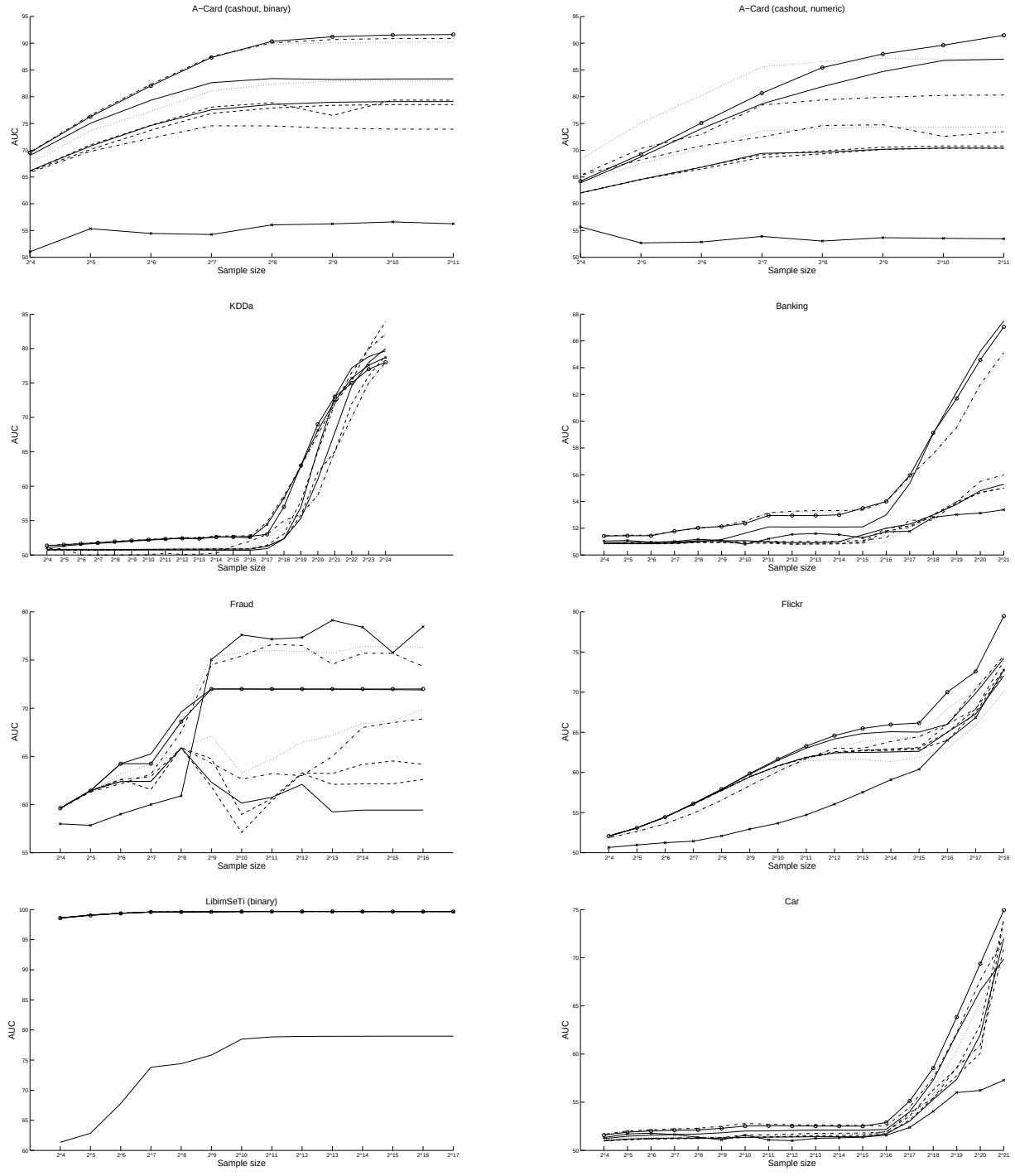

Figure 10: Learning curves in the feature dimension (intelligent feature selection).

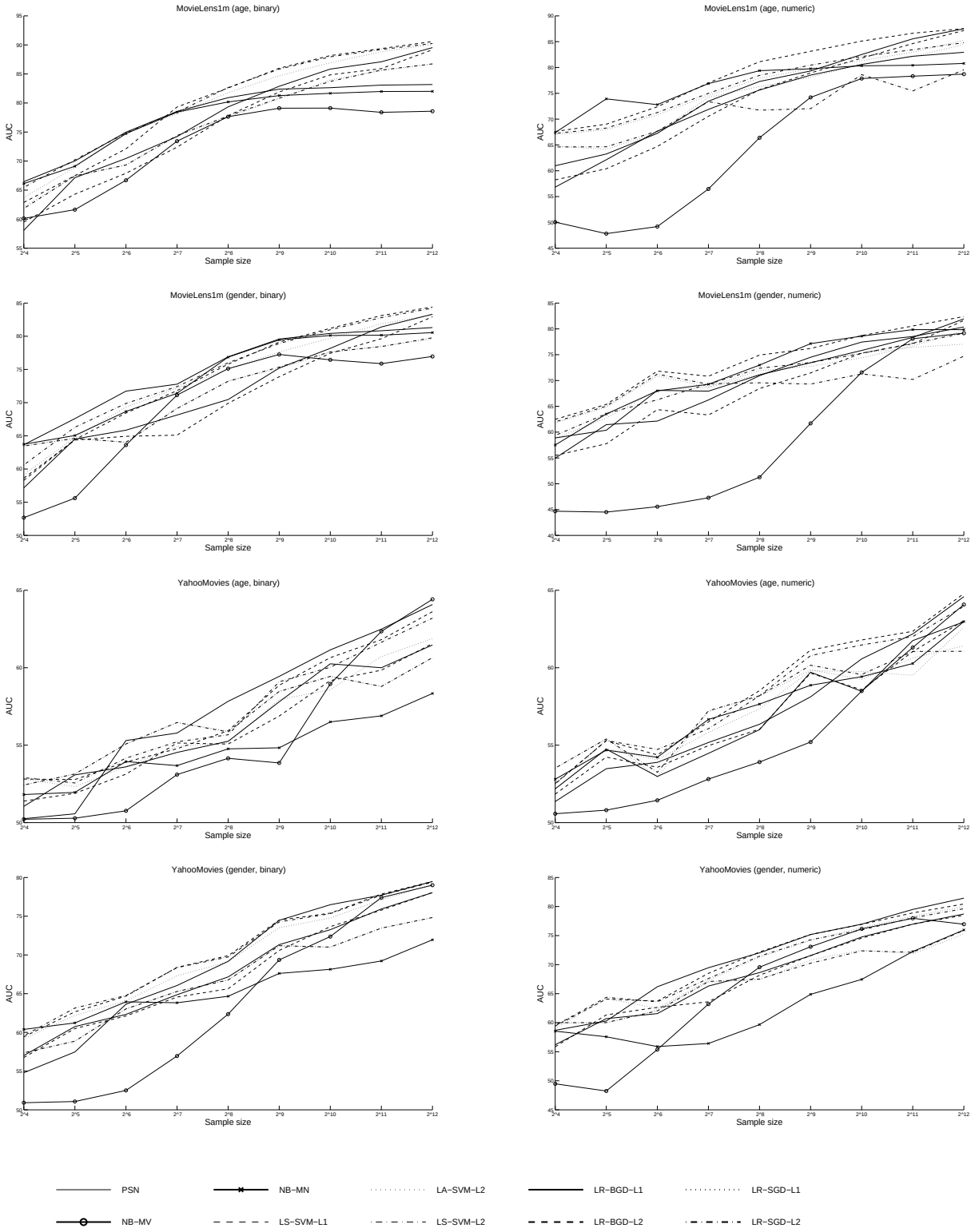

Figure 11: Learning curves in the instances dimension.

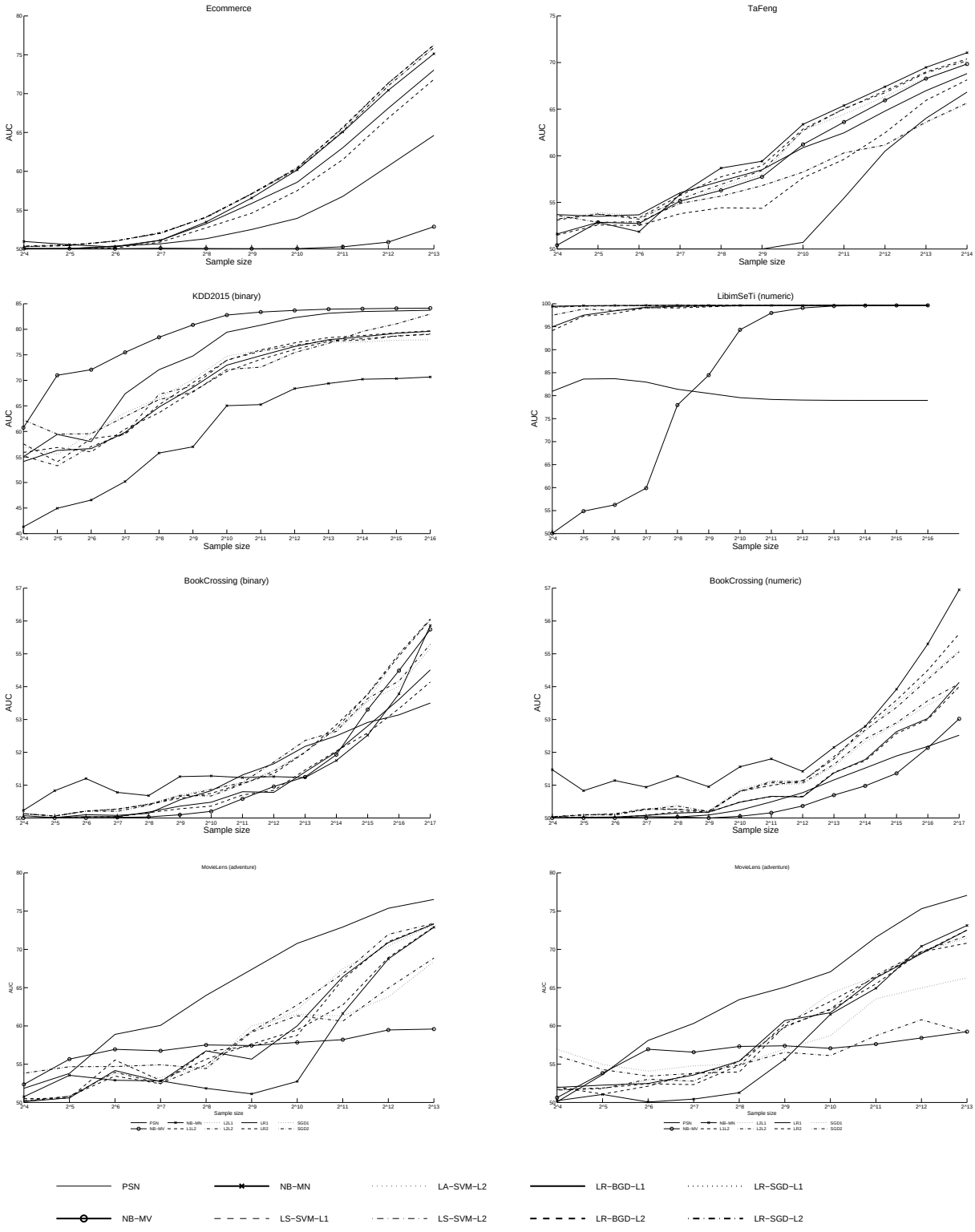

Figure 12: Learning curves in the instances dimension.

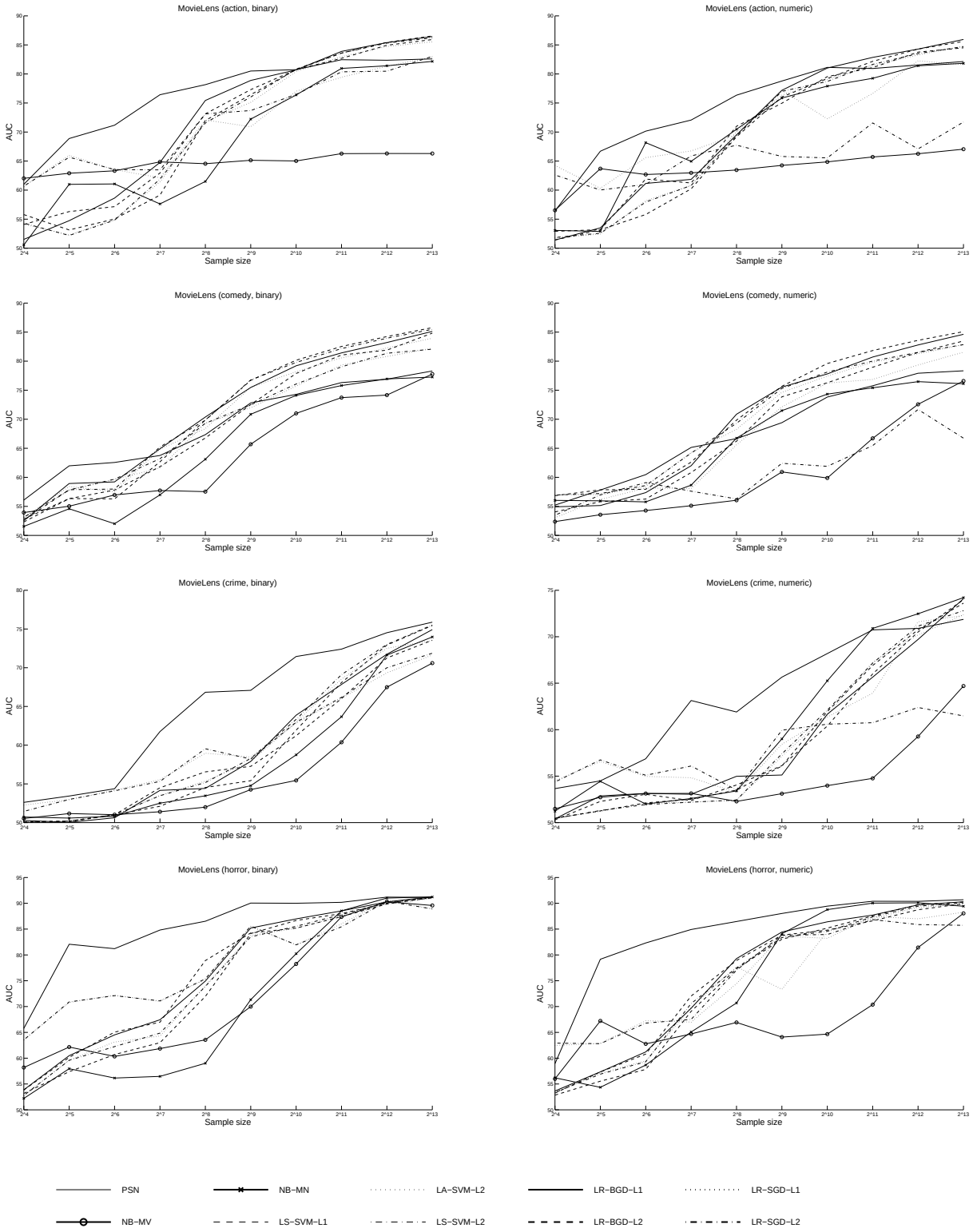

Figure 13: Learning curves in the instances dimension.

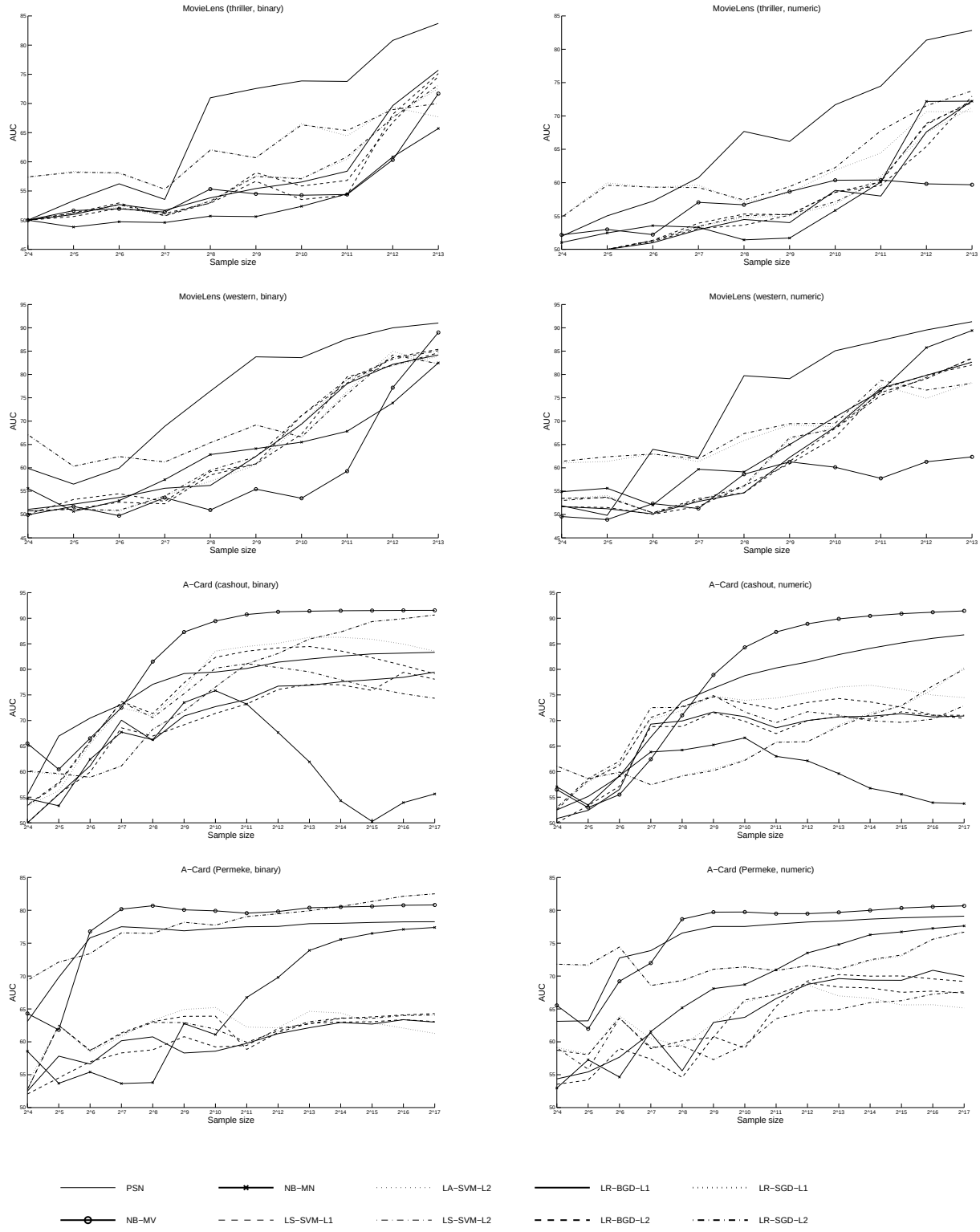

Figure 14: Learning curves in the instances dimension.

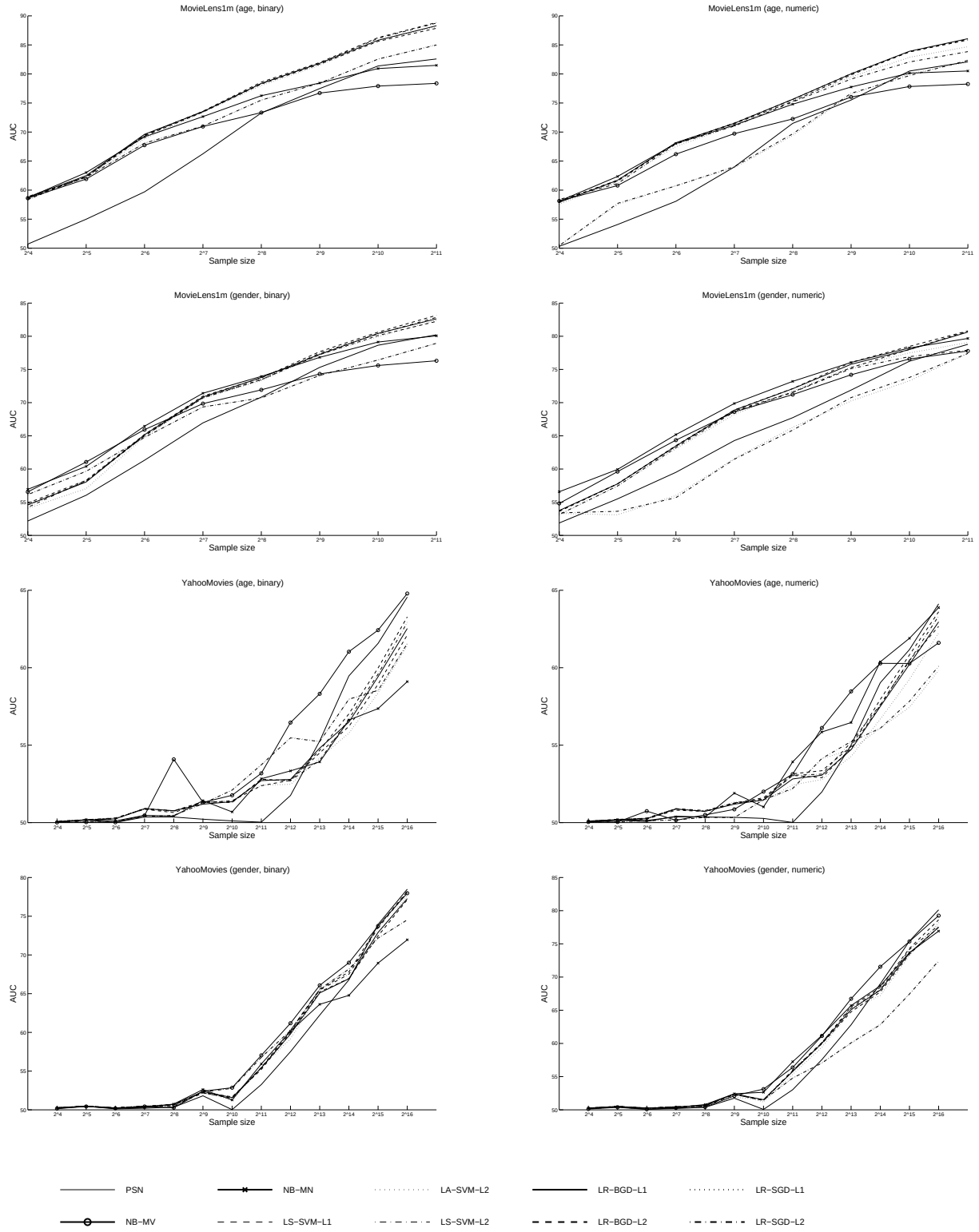

Figure 15: Learning curves in the features dimension (random feature selection).

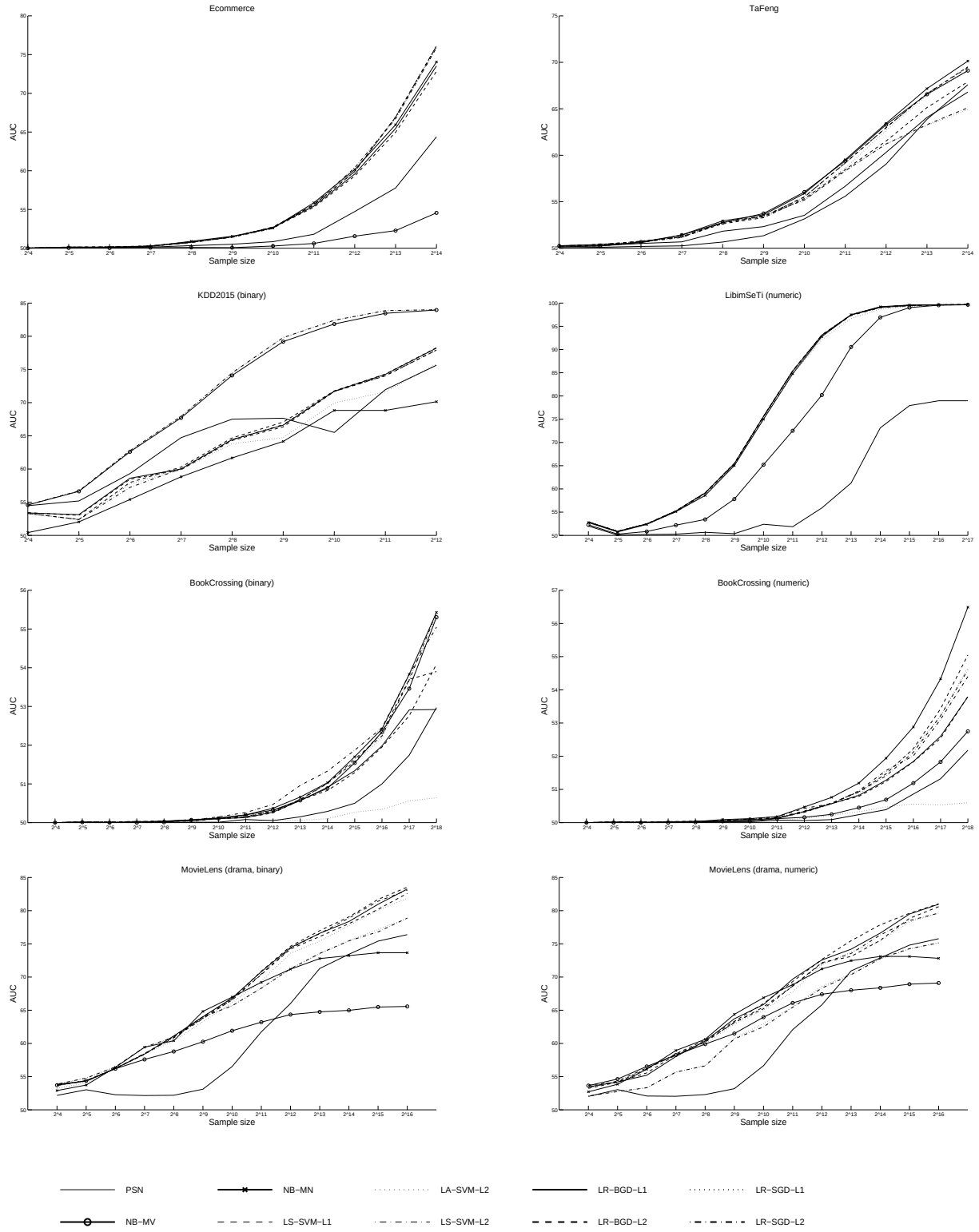

Figure 16: Learning curves in the features dimension (random feature selection).

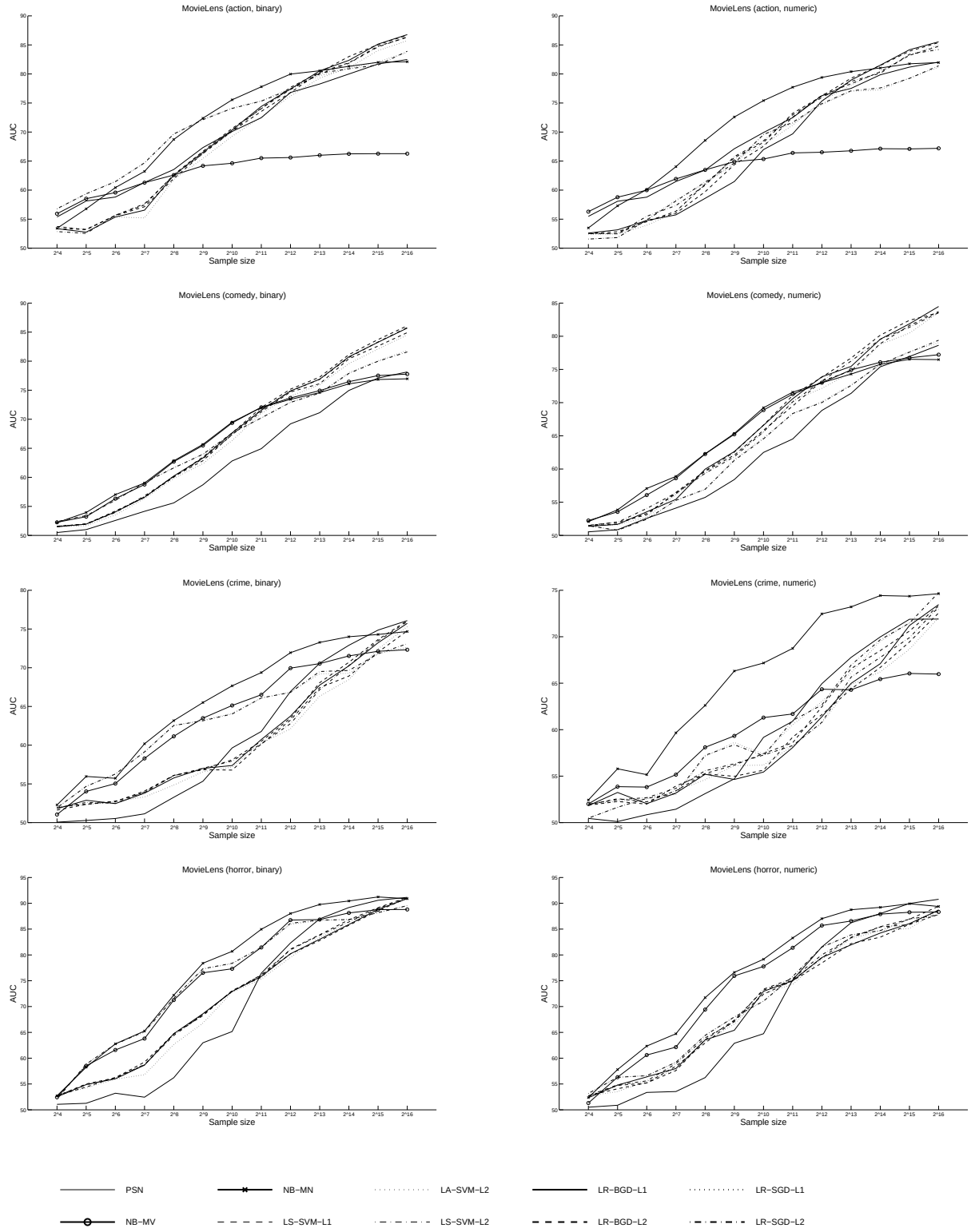

Figure 17: Learning curves in the features dimension (random feature selection).

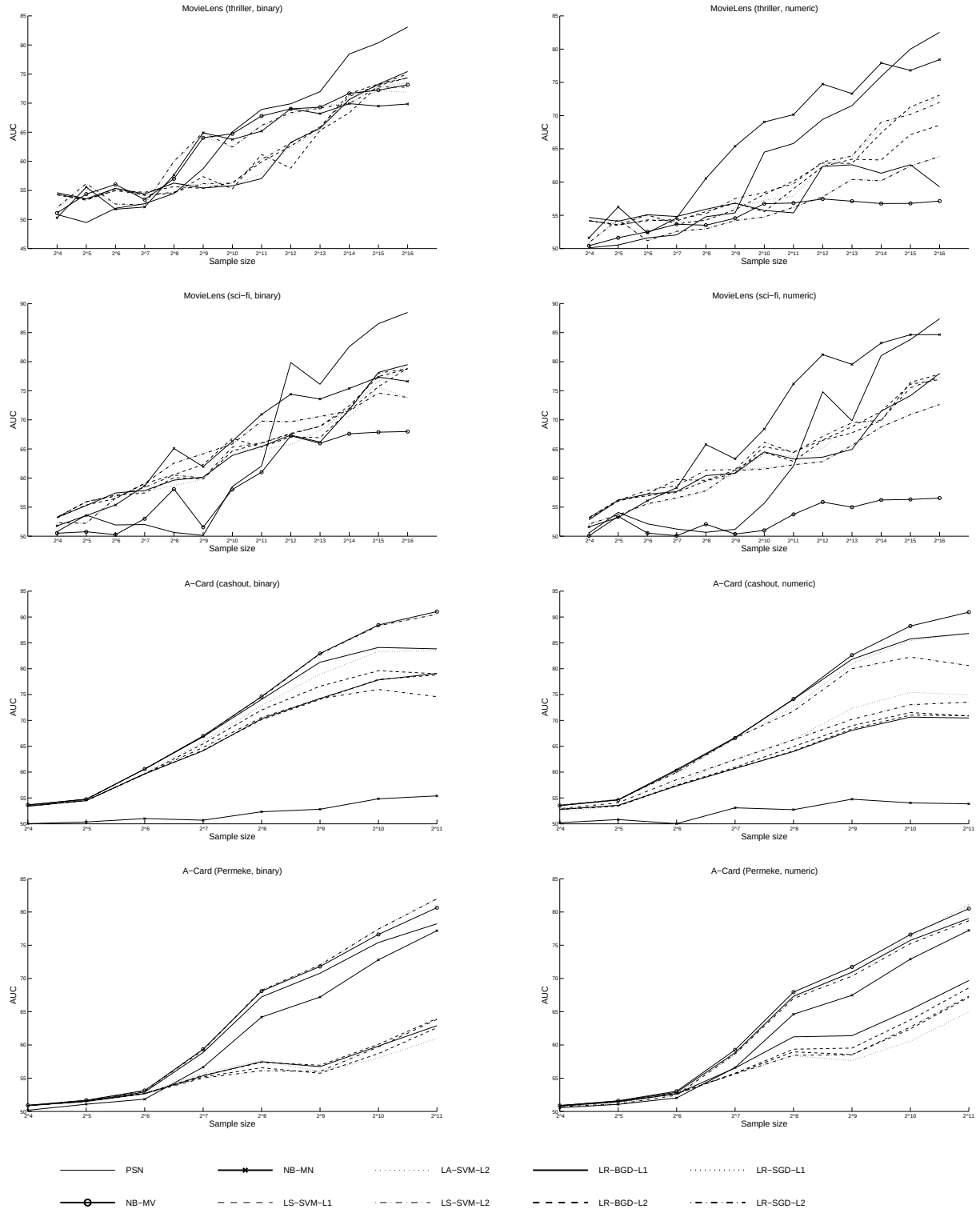

Figure 18: Learning curves in the features dimension (random feature selection).

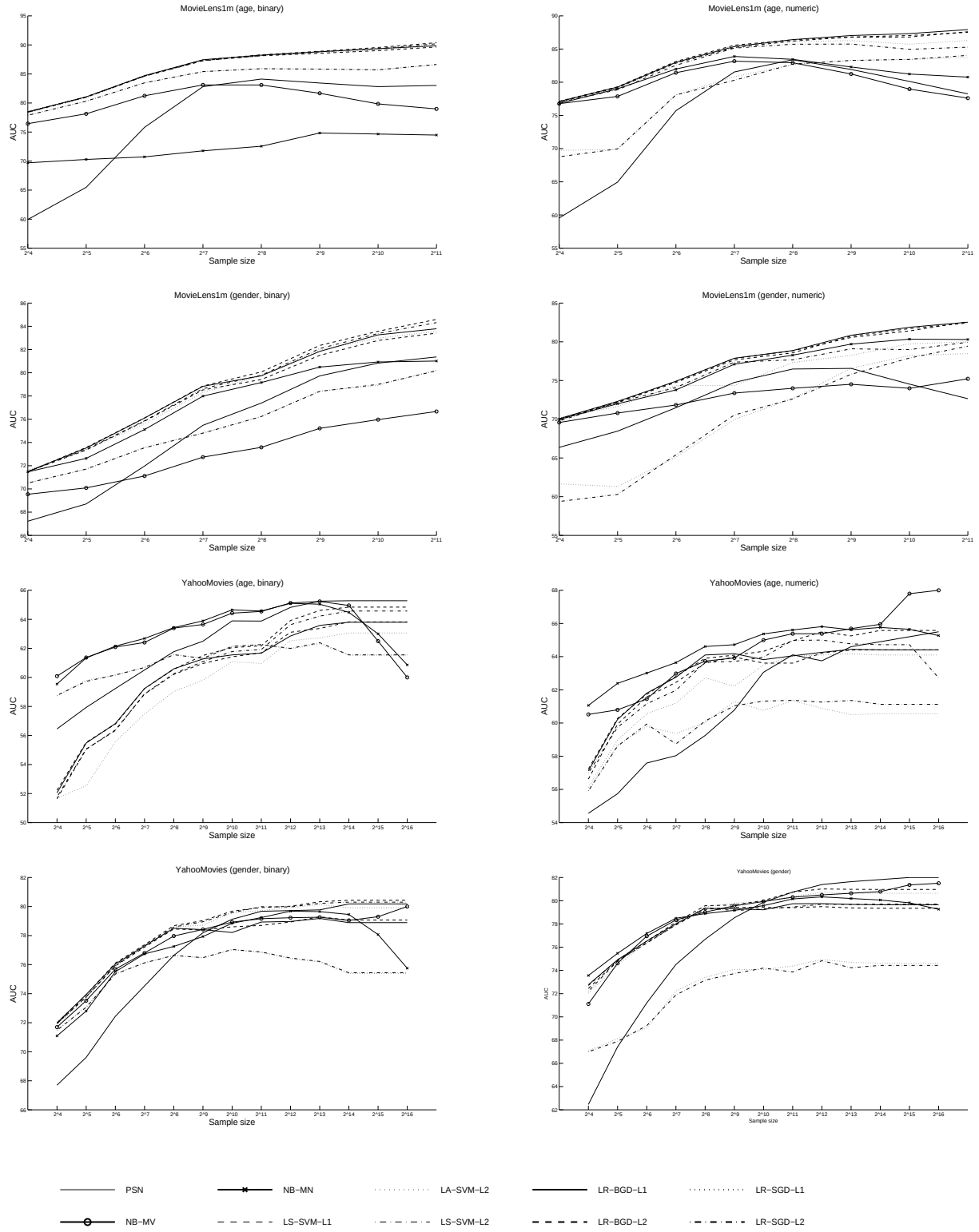

Figure 19: Learning curves in the features dimension (intelligent feature selection).

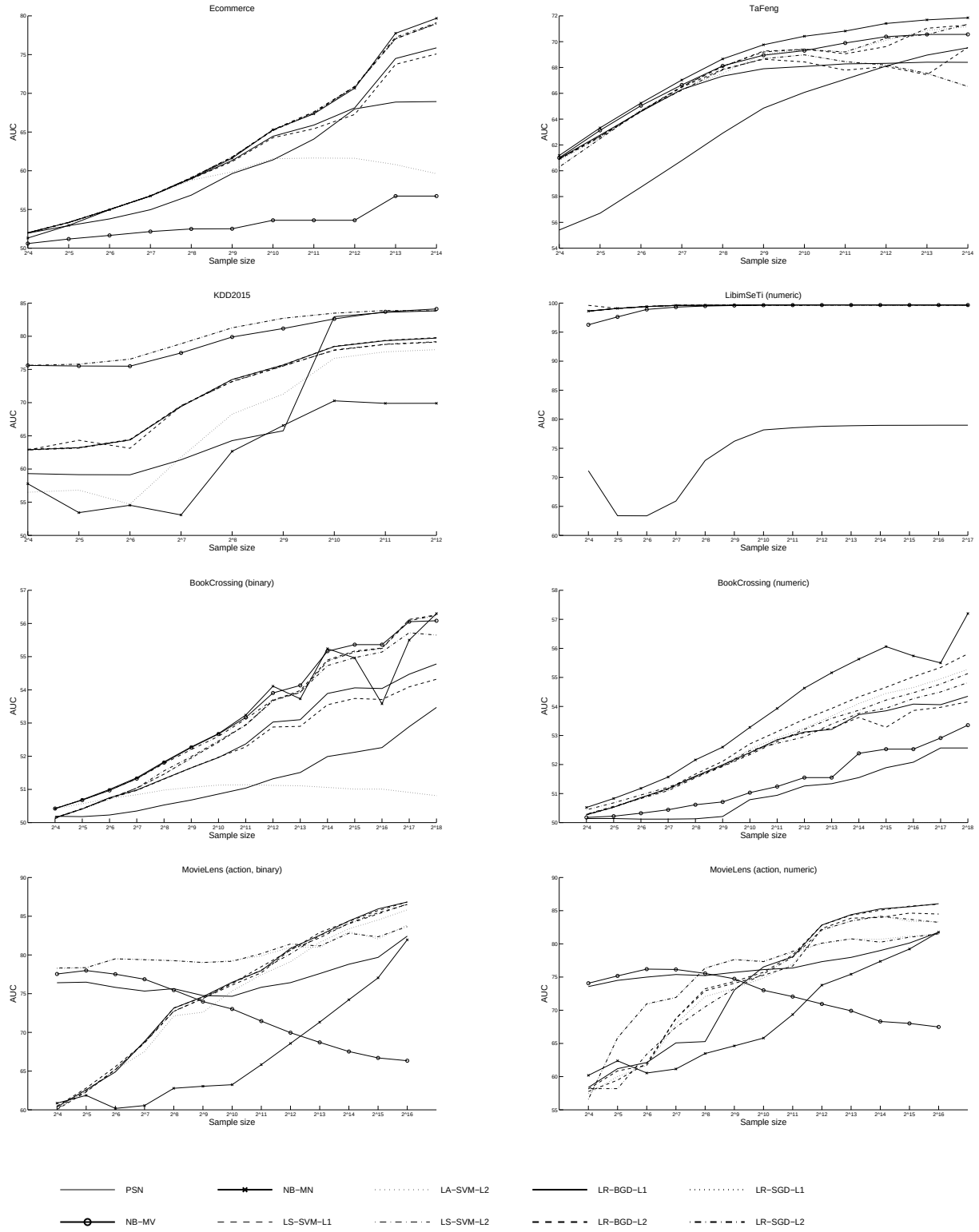

Figure 20: Learning curves in the features dimension (intelligent feature selection).

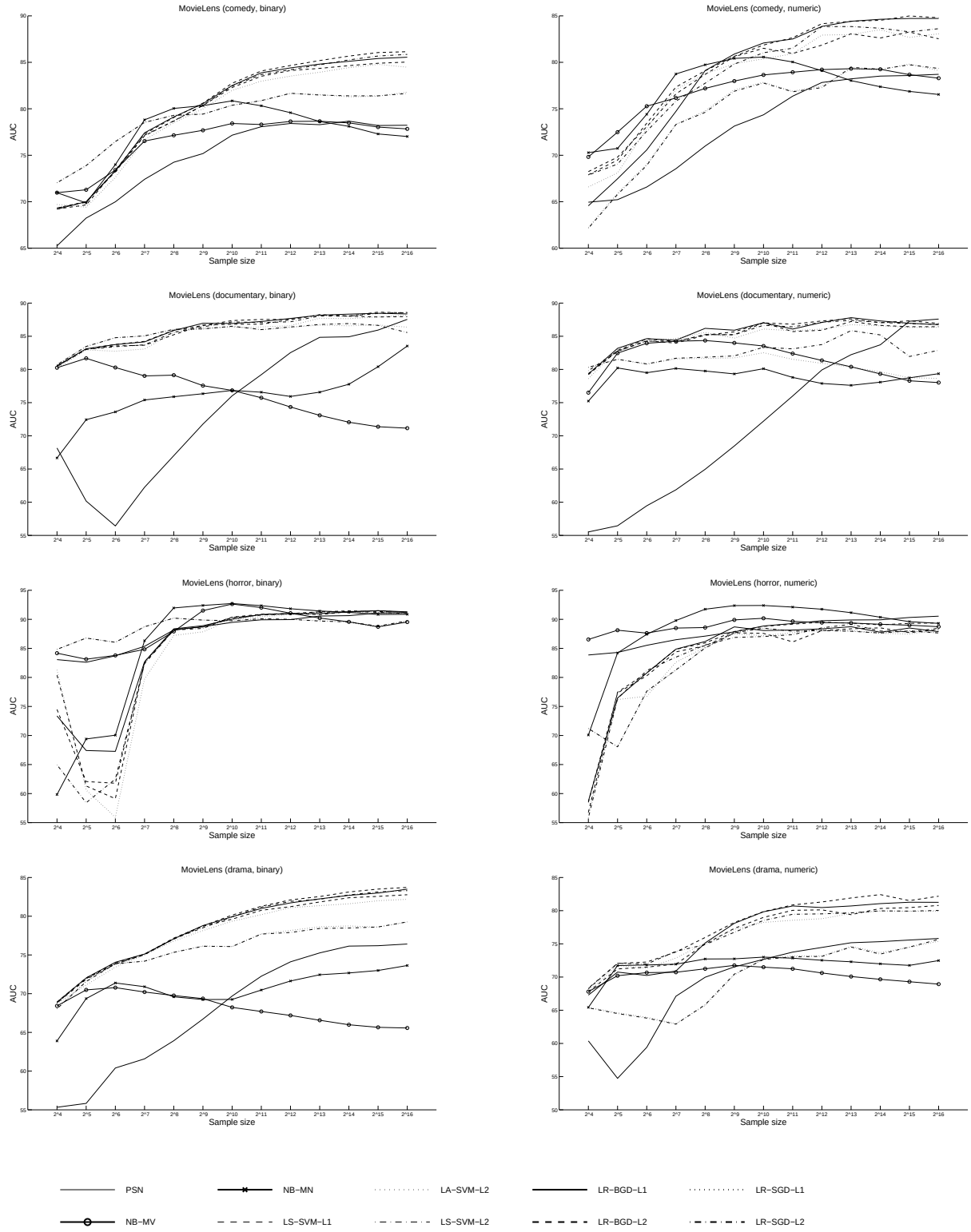

Figure 21: Learning curves in the features dimension (intelligent feature selection).

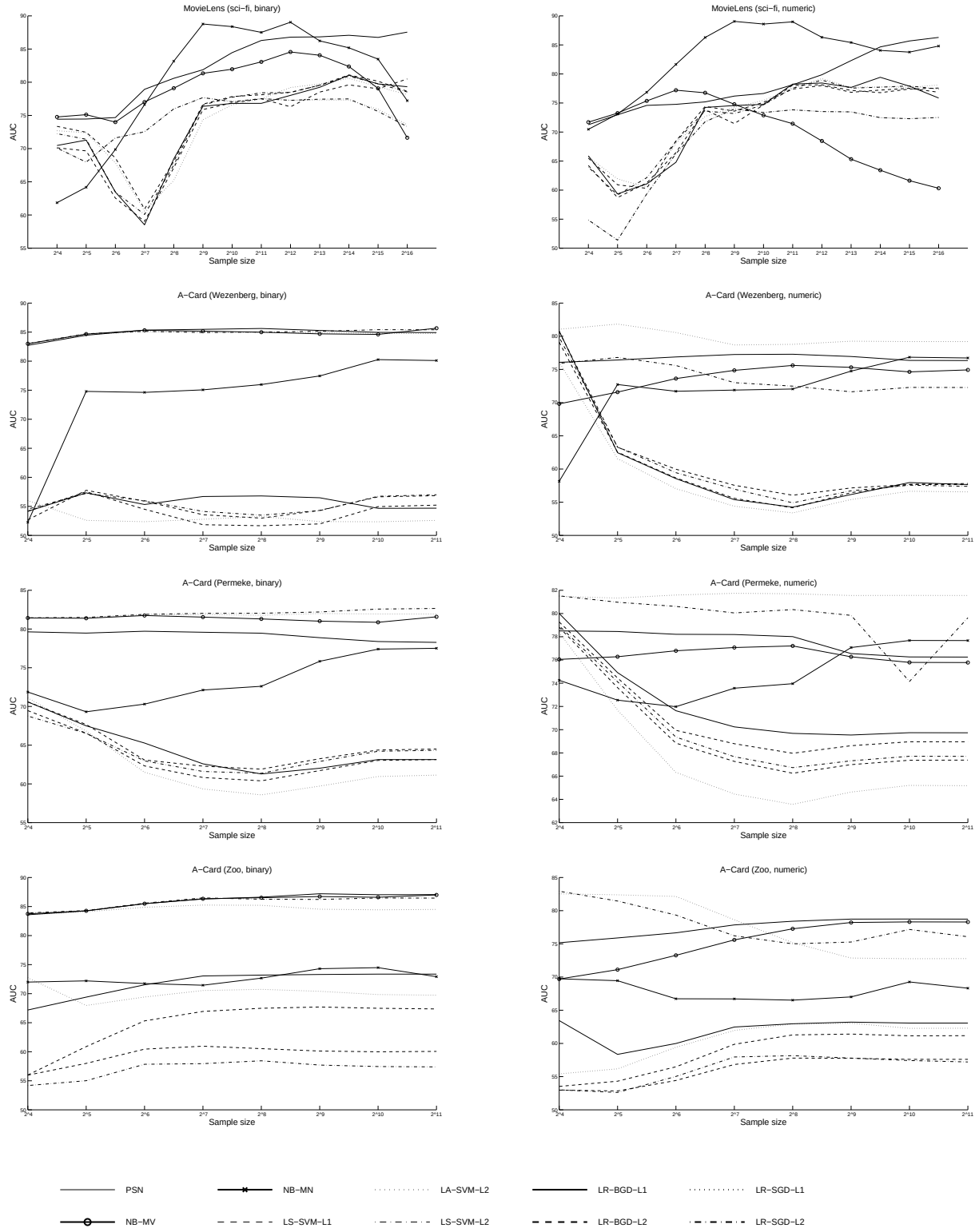

Figure 22: Learning curves in the features dimension (intelligent feature selection).

References

Agarwal A, Chapelle O, Dudík M, Langford J (2014) A reliable effective terascale linear learning system. Journal of Machine Learning Research 15:1111–1133

Bannur SN (2011) Detecting Malicious Webpages using Content Based Classication. Master's thesis, University of California, San Diego

Bennett J, Lanning S (2007) The netflix prize. In: Proceedings of 2007 KDD Cup and Workshop

Bermejo P, Gámez JA, Puerta JM (2014) Speeding up incremental wrapper feature subset selection with naive bayes classifier. Knowledge-Based Systems 55:140–147

Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: International Conference on Computational Statistics (COMPSTAT), Springer, pp 177–186

Brain D, Webb GI (2002) The need for low bias algorithms in classification learning from large data sets. In: Principles of Data Mining and Knowledge Discovery, Springer, pp 62–73

Brozovsky L, Petricek V (2007) Recommender system for online dating service. Znalosti Conference pp 1–12

Cao L (2010) In-depth behavior understanding and use: the behavior informatics approach. Information Science 180(17):3067–3085

Cha M, Mislove A, Gummadi KP (2009) A measurement-driven analysis of information propagation in the Flickr social network. In: International Conference on World Wide Web (WWW), ACM, pp 721–730

Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3)

Chang F, Guo CY, Lin XR, Lu CJ (2010) Tree decomposition for large-scale SVM problems. Journal of Machine Learning Research 11:2935–2972

Chen Y, Pavlov D, Canny JF (2009) Large-scale behavioral targeting. In: International Conference on Knowledge Discovery and Data Mining (SIGKDD), ACM, pp 209–218

Clark J, Provost F (2016) Matrix-factorization-based dimensionality reduction in the predictive modeling process: a design science perspective. Tech. rep., Department of Information, Operations, and Management Sciences, New York University, USA

Colas F, Brazdil P (2006) Comparison of svm and some older classification algorithms in text classification tasks. Artificial Intelligence in Theory and Practice pp 169–178

Collobert R, Sinz F, Weston J, Bottou L (2006) Large scale transductive SVMs. Journal of Machine Learning Research 7:1687–1712

Cortes C, Vapnik V (1995) Support-vector networks. Machine Learning 20(3):273–297

Dalessandro B (2013) Bring the noise: embracing randomness is the key to scaling up machine learning algorithms. Big Data 1(2):110–112

Dalessandro B, Chen D, Raeder T, Perlich C, Han Williams M, Provost F (2014) Scalable hands-free transfer learning for online advertising. In: International Conference on Knowledge Discovery and Data Mining (SIGKDD), ACM, pp 1573–1582

De Bock KW, Van den Poel D (2010) Predicting website audience demographics for web advertising targeting using multi-website clickstream data. Fundamenta Informaticae 98(1):49–70

De Cnudde S, Martens D (2015) Loyal to your city? A data mining analysis of a public service loyalty program. Decision Support Systems 73:74–84

De Cnudde S, Moeyersoms J, Stankova M, Tobback E, Javaly V, Martens D (2015) Who cares about your Facebook friends? Credit scoring for microfinance. Tech. rep., Department of Applied Economics, Antwerp University, Belgium

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7:1–30

Do TN, Lenca P, Lallich S, Pham NK (2009) Classifying very-high-dimensional data with random forests of oblique decision trees. In: EGC (best of volume), Springer, pp 39–55

Donoho DL (2000) High-dimensional data analysis: the curses and blessings of dimensionality. AMS Conference on Math Challenges of the 21st Century pp 1–32

Dumais S, Platt J, Heckerman D, Sahami M (1998) Inductive learning algorithms and representations for text categorization. In: Proceedings of the seventh international conference on Information and knowledge management, ACM, pp 148–155

Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) LIBLINEAR: a library for large linear classification. Journal of Machine Learning Research 9:1871–1874

Fawcett T (2006) An introduction to ROC analysis. Pattern Recognition Letters 27(8):861–874

Fawcett T, Provost F (1997) Adaptive fraud detection. Data Mining and Knowledge Discovery 1(3):291–316

Fernández-Delgado M, Cernadas E, Barro S, Amorim D (2014) Do we need hundreds of classifiers to solve real world classification problems? Journal of Machine Learning Research 15:3133–

3181

Forman G (2003) An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research 3:1289–1305

Forman G, Scholz M, Rajaram S (2009) Feature shaping for linear svm classifiers. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 299–308

Junqué de Fortuny E, Martens D, Provost F (2013) Predictive modeling with big data: is bigger really better? Big Data 1(4):215–226

Junque de Fortuny E, Martens D, Provost F (2013) Wallenius naive bayes

Junqué de Fortuny E, Stankova M, Moeyersoms J, Minnaert B, Provost F, Martens D (2014) Corporate residence fraud detection. In: International Conference on Knowledge Discovery and Data Mining (SIGKDD), ACM, pp 1650–1659

Junqué de Fortuny E, Evgeniou T, Martens D, Provost F (2015) Iteratively refining SVMs using priors. In: International Conference on Big Data (Big Data), IEEE, pp 46–52

Friedman JH (1997) On bias, variance, 0/1-loss, and the curse of dimensionality. Data Mining and Knowledge Discovery 1(1):55–77

Gigerenzer G, Goldstein DG (1996) Reasoning the fast and frugal way: models of bounded rationality. Psychological review 103(4):650

Gigerenzer G, Todd PM, ABC Research Group t, et al. (1999) Simple heuristics that make us smart. Oxford University Press

Goel S, Hofman JM, Sirer MI (2012) Who does what on the web: a large-scale study of browsing behavior. In: International Conference on Web and Social Media (ICWSM), AAAI

Green KC, Armstrong JS (2015) Simple versus complex forecasting: The evidence. Journal of Business Research 68(8):1678–1685

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. Journal of Machine Learning Research 3:1157–1182

Hand DJ, Yu K (2001) Idiot's Bayes—not so stupid after all? International Statistical Review 69(3):385–398

Heaps HS (1978) Information retrieval: Computational and theoretical aspects. Academic Press, Inc.

Hill S, Provost F, Volinsky C (2006) Network-based marketing: Identifying likely adopters via consumer networks. Statistical Science pp 256–276

Hsu CW, Chang CC, Lin CJ (2003) A practical guide to support vector classification. Tech. rep., National Taiwan University, Taipei, Taiwan

Hu X (2005) A data mining approach for retailing bank customer attrition analysis. Applied Intelligence 22(1):47–60

Huang HS, Lin KL, Hsu JYj, Hsu CN (2005) Item-triggered recommendation for identifying potential customers of cold sellers in supermarkets. In: Beyond Personalization Workshop on the Next Stage of Recommender Systems Research, pp 37–42

Huang J, Lu J, Ling CX (2003) Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. In: International Conference on Data Mining (ICDM), IEEE, pp 553–556

Iman RL, Davenport JM (1980) Approximations of the critical region of the Friedman statistic. Communications in Statistics—Theory and Methods 9(6):571–595

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

King RD, Feng C, Sutherland A (1995) Statlog: comparison of classification algorithms on large real-world problems. Applied Artificial Intelligence 9(3):289–333

Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence (IJCAI), vol 14, pp 1137–1145

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

Langford J, Li L, Strehl A (2007) Vowpal Wabbit online learning project. Tech. rep., `http://hunch.net`

Langley P, Iba W, Thompson K (1992) An analysis of Bayesian classifiers. In: National Conference on Artificial Intelligence, AAAI, vol 90, pp 223–228

Li K, Du TC (2012) Building a targeted mobile advertising system for location-based services. Decision Support Systems 54(1):1–8

Li P, Owen A, Zhang CH (2012) One permutation hashing. In: Advances in Neural Information Processing Systems, pp 3113–3121

Li X, Wang H, Gu B, Ling CX (2015) Data sparseness in linear SVM. In: International Conference on Artificial Intelligence, AAAI, pp 3628–3634

Lim TS, Loh WY, Shih YS (2000) A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning 40(3):203–228

Lin CJ, Weng RC, Keerthi S (2008) Trust region Newton method for logistic regression. Journal of Machine Learning Research 9:627–650

Liu A, Ghosh J, Martin C (2007) Generative oversampling for mining imbalanced datasets. In: International Conference on Data Mining (ICDM), IEEE, pp 66–72

Liu J, Dolan P, Pedersen ER (2010) Personalized news recommendation based on click behavior. In: International Conference on Intelligent User Interfaces (IUI), ACM, pp 31–40

Macià N, Bernadó-Mansilla E (2014) Towards UCI+: a mindful repository design. Information Sciences 261(1):237–262

Macià N, Bernadó-Mansilla E, Orriols-Puig A, Ho TK (2013) Learner excellence biased by data set selection: A case for data characterisation and artificial data sets. Pattern Recognition 46(3):1054–1066

Macskassy SA, Provost F (2007) Classification in networked data: a toolkit and a univariate case study. Journal of Machine Learning Research 8:935–983

Martens D, Provost F (2014) Explaining data-driven document classifications. MIS Quarterly 38(1):73–100

Martens D, Provost F, Clark J, Junqué de Fortuny E (2016) Mining massive fine-grained behavior data to improve predictive analytics. Management Information Systems Quarterly (MISQ) 40(4)

McCallum A, Nigam K (1998) A comparison of event models for naive Bayes text classification. In: Workshop on Learning for Text Categorization, AAAI, pp 41–48

Metsis V, Androutsopoulos I, Paliouras G (2006) Spam filtering with naive bayes-which naive bayes? In: CEAS, vol 17, pp 28–69

Meyer D, Leisch F, Hornik K (2002) Benchmarking support vector machines. Tech. rep., Adaptive Information Systems and Modelling in Economics and Management Science, WU Vienna University of Economics and Business Administration, Austria

Michie D, Spiegelhalter DJ, Taylor CC (2009) Machine Learning, Neural and Statistical Classification. Overseas Press

Ng AY (2004) Feature selection, L1 vs. L2 regularization, and rotational invariance. In: International Conference on Machine Learning (ICML), ACM

Ng AY, Jordan A (2002) On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. Advances in Neural Information Processing Systems (NIPS) 14:841

Nie F, Huang Y, Wang X, Huang H (2014) New primal SVM solver with linear computational cost for big data classifications. In: International Conference on Machine Learning (ICML), ACM

Pandey S, Aly M, Bagherjeiran A, Hatch A, Ciccolo P, Ratnaparkhi A, Zinkevich M (2011) Learning to target: what works for behavioral targeting. In: International Conference on Information and Knowledge Management (CIKM), ACM, pp 1805–1814

Perlich C, Provost F, Simonoff JS (2003) Tree induction vs. logistic regression: a learning-curve analysis. Journal of Machine Learning Research 4:211–255

Perlich C, Dalessandro B, Raeder T, Stitelman O, Provost F (2014) Machine learning for targeted display advertising: transfer learning in action. Machine Learning 95(1):103–127

Provost F, Fawcett T (2013) Data Science for Business: What You Need to Know about Data Mining and Data-analytic Thinking. O'Reilly Media, Inc.

Provost F, Kolluri V (1999) A survey of methods for scaling up inductive algorithms. Data Mining and Knowledge Discovery 3(2):131–169

Provost F, Fawcett T, Kohavi R (1998) The case against accuracy estimation for comparing induction algorithms. In: International Conference on Machine Learning (ICML), ACM, pp 445–453

Ralaivola L, d'Alché Buc F (2001) Incremental support vector machine learning: a local approach. In: International Conference on Artificial Neural Networks (ICANN), Springer, pp 322–330

Schneider KM (2004) On word frequency information and negative evidence in naive bayes text classification. EsTAL 3230:474–486

Sebastiani F (2002) Machine learning in automated text categorization. ACM computing surveys (CSUR) 34(1):1–47

Shavlik JW, Mooney RJ, Towell GG (1991) Symbolic and neural learning algorithms: an experimental comparison. Machine Learning 6(2):111–143

Shmueli G (2016) Analyzing behavioral big data: methodological, practical, ethical, and moral issues. Quality Engineering 29:57–74

Sigurd B, Eeg-Olofsson M, Van Weijer J (2004) Word length, sentence length and frequency–zipf revisited. Studia Linguistica 58(1):37–52

Stankova M, Martens D, Provost F (2014) Classification over bipartite graphs through projection. Tech. rep., Department of Applied Economics, Antwerp University, Belgium

Tan M, Tsang IW, Wang L (2014) Towards ultrahigh dimensional feature selection for big data. Journal of Machine Learning Research 15:1371–1429

Tsang IW, Kwok JT, Cheung PM (2005) Core vector machines: fast SVM training on very large data sets. Journal of Machine Learning Research

6:363–392

Verbeke W, Martens D, Baesens B (2014) Social network analysis for customer churn prediction. Applied Soft Computing 14(3):431–446

Walker T (2016) So much data, so little time: Using sequential data analysis to monitor behavioral changes. MethodsX 3:560–568

Wallace BC, Small K, Brodley CE, Trikalinos TA (2011) Class imbalance, redux. In: International Conference on Data Mining (ICDM), IEEE, pp 754–763

Weinberger K, Dasgupta A, Langford J, Smola A, Attenberg J (2009) Feature hashing for large scale multitask learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, pp 1113–1120

Wolpert DH (1996) The lack of a priori distinctions between learning algorithms. Neural computation 8(7):1341–1390

Wu X, Zhu X, Wu GQ, Ding W (2014) Data mining with big data. Transactions on Knowledge and Data Engineering 26(1):97–107

Yang Q, Wu X (2006) 10 challenging problems in data mining research. International Journal of Information Technology & Decision Making 5(4):597–604

Yu HF, Lo HY, Hsieh HP, Lou JK, McKenzie TG, Chou JW, Chung PH, Ho CH, Chang CF, Wei YH, et al. (2010) Feature engineering and classifier ensemble for KDD Cup 2010. In: International Conference on Knowledge Discovery and Data Mining KDD Cup 2010 Workshop (SIGKDD), ACM

Zhu J, Rosset S, Hastie T, Tibshirani R (2003) 1-norm support vector machines. Advances in Neural Information Processing Systems (NIPS) 16(1):49–56

Ziegler CN, McNee SM, Konstan JA, Lausen G (2005) Improving recommendation lists through topic diversification. In: International Conference on World Wide Web (WWW), ACM, pp 22–32

Zipf GK (2016) Human behavior and the principle of least effort: An introduction to human ecology. Ravenio Books