JMLR: Workshop and Conference Proceedings 4: 90-105 New challenges for feature selection

On the Relationship Between Feature Selection and Classification Accuracy

Andreas G. K. Janecek ANDREAS.JANECEK@UNIVIE.AC.AT
Wilfried N. Gansterer WILFRIED.GANSTERER@UNIVIE.AC.AT

University of Vienna, Research Lab Computational Technologies and Applications
Lenaugasse 2/8, 1080 Vienna, Austria

Michael A. Demel MICHAEL.DEMEL@UNIVIE.AC.AT
Gerhard F. Ecker GERHARD.F.ECKER@UNIVIE.AC.AT

University of Vienna, Emerging Field Pharmacoinformatics, Department of Medicinal Chemistry,
Althanstrasse 14, 1090 Vienna, Austria

Editor: Saeys et al.

Abstract

Dimensionality reduction and feature subset selection are two techniques for reducing the attribute space of a feature set, which is an important component of both supervised and unsupervised classification or regression problems. While in feature subset selection a subset of the original attributes is extracted, dimensionality reduction in general produces linear combinations of the original attribute set.

In this paper we investigate the relationship between several attribute space reduction techniques and the resulting classification accuracy for two very different application areas. On the one hand, we consider e-mail filtering, where the feature space contains various properties of e-mail messages, and on the other hand, we consider drug discovery problems, where quantitative representations of molecular structures are encoded in terms of information-preserving descriptor values.

Subsets of the original attributes constructed by filter and wrapper techniques as well as subsets of linear combinations of the original attributes constructed by three different variants of the principle component analysis (PCA) are compared in terms of the classification performance achieved with various machine learning algorithms as well as in terms of runtime performance. We successively reduce the size of the attribute sets and investigate the changes in the classification results. Moreover, we explore the relationship between the variance captured in the linear combinations within PCA and the resulting classification accuracy.

The results show that the classification accuracy based on PCA is highly sensitive to the type of data and that the variance captured the principal components is not necessarily a vital indicator for the classification performance.

1. Introduction and Related Work

As the dimensionality of the data increases, many types of data analysis and classification problems become significantly harder. Sometimes the data also becomes increasingly sparse in the space it occupies. This can lead to big problems for both supervised and unsupervised learning. In the literature, this phenomenon is referred to as the *curse of dimensionality* (Powell, 2007). On the one hand, in the case of supervised learning or classification the available training data may be too small, i. e, there may be too few data objects to allow the creation of a reliable model for assigning a class to all possible objects. On the other hand, for unsupervised learning methods or clustering algorithms, various vitally important definitions like density or distance between points may become less convincing (as more dimensions tend to make the proximity between points more uniform). As a result, a high number of features can lead to lower classification accuracy and clusters of poor quality. High dimensional data is also a serious problem for many classification algorithms due to its high computational cost and memory usage. Besides this key factor, Tan et al. (2005) also mention that a reduction of the attribute space leads to a better understandable model and simplifies the usage of different visualization techniques. Several extensive surveys of various feature selection and dimensionality reduction approaches can be found in the literature, for example, in Molina et al. (2002) or Guyon and Elisseeff (2003).

Principle component analysis (PCA) is a well known data preprocessing technique to capture linear dependencies among attributes of a data set. It compresses the attribute space by identifying the strongest patterns in the data, i. e., the attribute space is reduced by the smallest possible amount of information about the original data.

Howley et al. (2006) have investigated the effect of PCA on machine learning accuracy with high dimensional spectral data based on different pre-processing steps. They use the NIPALS method (Geladi and Kowalski, 1986) to iteratively compute only the first n principle components (PCs) of a data sample until a required number of PCs have been generated. Their results show that using this PCA method in combination with classification may improve the classification accuracy when dealing with high dimensional data. For carefully selected pre-processing techniques, the authors show that the addition of the PCA step results in either the same error (for a C4.5 and a RIPPER classifier) or a numerically smaller error (linear SVM, RBF SVM, k-NN and linear regression). Popelinsky (2001) has analyzed the effect of PCA on three different machine learning methods (C5.0, instance-based learner and naive Bayes). In one test-run, the first n PCs (i.e., linear combinations of the original attributes) were added to the original attributes, in the second test run, the principle components replaced them. The results show that adding the PCs to the original attributes slightly improved the classification accuracy for all machine learning algorithms (mostly for small numbers of n), whereas replacing the original attributes only increased the accuracy for one algorithm (naive Bayes). Gansterer et al. (2008) have investigated the benefits of dimensionality reduction in the context of latent semantic indexing for e-mail spam detection.

Different techniques can be applied to perform a principle component analysis, for example, either the covariance or the correlation matrix can be used to calculate the eigenvalue decomposition. Moreover, scaling of the original data may have a strong influence on the PCs. Attributes resulting from these different PCA variants differ significantly in their cov-

erage of the variability of the original attributes. To the best of our knowledge no systematic studies have been carried out to explore the relationship between the variability captured in the PCs used and the accuracy of machine learning algorithms operating on them. One of the objectives of this paper is to summarize investigations of this issue. More generally, we investigate the variation of the classification accuracy depending on the choice of the feature set (that includes the choice of specific variants for calculating the PCA) for two very different data sets. Another important aspect motivating such investigations are questions relating to how the classification accuracy based on PCA subsets compares to classification accuracy based on subsets of the original features of the same size, or how to identify the smallest subset of original features which yields a classification accuracy comparable to the one of a given PCA subset.

2. Feature Reduction

In the following we distinguish two classes of feature reduction strategies: Feature subset selection (FS) and dimensionality reduction (DR). The main idea of feature subset selection is to remove redundant or irrelevant features from the data set as they can lead to a reduction of the classification accuracy or clustering quality and to an unnecessary increase of computational cost (Blum and Langley, 1997), (Koller and Sahami, 1996). The advantage of FS is that no information about the importance of single features is lost. On the other hand, if a small set of features is required and the original features are very diverse, information may be lost as some of the features must be omitted. With dimensionality reduction techniques the size of the attribute space can often be decreased strikingly without loosing a lot of information of the original attribute space. An important disadvantage of DR is the fact that the linear combinations of the original features are usually not interpretable and the information about how much an original attribute contributes is often lost.

2.1 Feature (Subset) Selection

Generally speaking, there are three types of feature subset selection approaches: filters, wrappers, and embedded approaches which perform the features selection process as an integral part of a machine learning (ML) algorithm.

Filters are classifier agnostic pre-selection methods which are independent of the later applied machine learning algorithm. Besides some statistical filtering methods like Fisher score (Furey et al., 2000) or Pearson correlation (Miyahara and Pazzani, 2000), *information gain*, originally used to compute splitting criteria for decision trees, is often used to find out how well each single feature separates the given data set.

The overall entropy I of a given dataset S is defined as

$$I(S) := -\sum_{i=1}^{C} p_i \log_2 p_i,$$

where C denotes the total number of classes and p_i the portion of instances that belong to class i. The reduction in entropy or the *information gain* is computed for each attribute A according to

$$IG(S, A) = I(S) - \sum_{v \epsilon A} \frac{|S_{A,v}|}{|S|} I(S_{A,v}),$$

where v is a value of A and $S_{A,v}$ is the set of instances where A has value v.

Wrappers are feedback methods which incorporate the ML algorithm in the FS process, i.e., they rely on the performance of a specific classifier to evaluate the quality of a set of features. Wrapper methods search through the space of feature subsets and calculate the estimated accuracy of a single learning algorithm for each feature that can be added to or removed from the feature subset. The feature space can be searched with various strategies, e. g., forwards (i. e., by adding attributes to an initially empty set of attributes) or backwards (i. e., by starting with the full set and deleting attributes one at a time). Usually an exhaustive search is too expensive, and thus non-exhaustive, heuristic search techniques like genetic algorithms, greedy stepwise, best first or random search are often used (see, for details, Kohavi and John (1997)).

Filters vs. wrappers. In the filter approach the FS is independent of a machine learning algorithm (classifier). This is computationally more efficient but ignores the fact that the selection of features may depend on the learning algorithm. On the other hand, the wrapper method is computationally more demanding, but takes dependencies of the feature subset on the learning algorithm into account.

2.2 Dimensionality Reduction

Dimensionality reduction (DR) refers to algorithms and techniques which create new attributes as combinations of the original attributes in order to reduce the dimensionality of a data set (Liu and Motoda, 1998). The most important DR technique is the principal component analysis (PCA), which produces new attributes as linear combinations of the original variables. In contrast, the goal of a factor analysis (Gorsuch, 1983) is to express the original attributes as linear combinations of a small number of hidden or latent attributes. The factor analysis searches for underlying (i. e. hidden or latent) attributes that summarize a group of highly correlated attributes.

PCA. The goal of PCA (Jolliffe, 2002) is to find a set of new attributes (PCs) which meets the following criteria: The PCs are (*i*) linear combinations of the original attributes, (*ii*) orthogonal to each other, and (*iii*) capture the maximum amount of variation in the data. Often the variability of the data can be captured by a relatively small number of PCs, and, as a result, PCA can achieve high dimensionality reduction with usually lower noise than the original patterns. The principle components are not always easy to interpret, and, in addition to that, PCA depends on the scaling of the data.

Mathematical background. The *covariance* of two attributes is a measure how strongly the attributes vary together. The covariance of two random variables x and y of a sample with size n and mean $\overline{x}$, $\overline{y}$ can be calculated as

$$\mathrm{Cov}(x, y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})(y - \overline{y}).$$

When x and y are normalized by their standard deviations σ_x and σ_y, then the covariance of x and y is equal to the correlation coefficient of x and y, $\text{Corr}(x, y) = \text{Cov}(x, y)/\sigma_x\sigma_y$, which indicates the strength *and* direction of a linear relationship between x and y.
Given an m by n matrix D, whose m rows are data objects and whose n columns are attributes, we can calculate the covariance matrix $\text{Cov}(D)$ which is constructed of the single covariances. If we shift the values of each attribute of D such that the mean of each attribute is 0, then $\text{Cov}(D) = D^T D$.

Tan et al. (2005) summarize four main properties of the PCA: (*i*) Each pair of attributes has covariance 0, (*ii*) the attributes are ordered descendingly with respect of their variance, (*iii*) the first attribute captures as much of the variance of the data as possible, and, (*iv*) each successive attribute captures as much of the remaining data as possible.

One way to obtain a transformation of the data which has these properties is based on the eigenvalue analysis of the covariance matrix. Let $\lambda_1, ..., \lambda_n$ be the non-negative descendingly ordered eigenvalues and $U = [\mathbf{u}_1, ..., \mathbf{u}_n]$ the matrix of eigenvectors of $\text{Cov}(D)$ (the i^{th} eigenvector corresponds to the i^{th} largest eigenvalue). The matrix $X = DU$ is the transformed data that satisfies the conditions mentioned above, where each attribute is a linear combination of the original attributes, the variance of the i^{th} new attribute is λ_i, and the sum of the variance of all new attributes is equal to the sum of the variances of the original attributes. The eigenvectors of $\text{Cov}(D)$ define a new set of orthogonal axes that can be viewed as a rotation of the original axes. The total variability of the data is still preserved, but the new attributes are now uncorrelated.

3. Experimental Evaluation

For the experimental evaluation we used MATLAB to compute three different variants of the PCA (cf. Section 3.2), and the WEKA toolkit (Witten and Frank, 2005) to compute the feature selection subsets (information gain and wrapper approach) and to measure the classification performance of the learning methods on each of these feature sets.

3.1 Data Sets

The data sets used for the experiments come from two completely different application areas and differ strongly in the number of instances and features and in their characteristics.

E-Mail data. The first data set consists of 10 000 e-mail messages (half of them spam, half of them not spam) taken from the TREC 2005 e-mail corpus (Cormack and Lynam, 2005). The values of the features for each message were extracted using the state-of-the-art spam filtering system SpamAssassin (SA) (Apache Software Foundation, 2006), where different parts of each e-mail message are checked by various tests, and each test assigns a certain value to each feature (positive values indicating spam messages, negative values indicating non-spam messages). Although the number of features determined by SA is rather large, only a small number of these features provide useful information. For the data set used only 230 out of 800 tests triggered at least once, resulting in a $10\,000 \times 230$ matrix.

Drug discovery data. The second data set comes from medicinal chemistry. The goal is to identify potential safety risks in an early phase of the drug discovery process in order to avoid costly and elaborate late stage failures in clinical studies (Ecker, 2005). This data set consists of 249 structurally diverse chemical compounds. 110 of them are

known to be substrates of P-glycoprotein, a macromolecule which is notorious for its potential to decrease the efficacy of drugs ("antitarget"). The remaining 139 compounds are non-substrates. The chemical structures of these compounds are encoded in terms of 366 information preserving descriptor values (features), which can be categorized into various groups, like simple physicochemical properties (e.,g., molecular weight), atomic properties (e. g. electronegativity), atom-type counts (e. g., number of oxygens), autocorrelation descriptors, and, additionally, "in-house" similarity-based descriptors (Zdrazil et al., 2007). Hence, our drug discovery (DD) data set is a 249×366 matrix.

Data characteristics. Whereas the e-mail data set is very sparse (97.5% of all entries are zero) the drug discovery data set contains only about 18% zero entries. Moreover, most of the e-mail features have the property that they are either zero or have a fixed value (depending on whether a test triggers or not). This is completely different from the drug discovery data set where the attribute values vary a lot (the range can vary from descriptors represented by small descrete numbers to descriptors represented by floating values having a theoretically infinite range).

Classification problem. In general, the performance of a binary classification process can be evaluated by the following quantities: True positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). In the context of the e-mail data set, a "positive" denotes an e-mail message which is classified as spam and a "negative" denotes an e-mail message which is classified as ham. Consequently, a "true positive" is a spam message which was (correctly) classified as spam, and a "false positive" is a ham message which was (wrongly) classified as spam.

In the context of our drug discovery data, "positives" are compounds which have a particular pharmaceutical activity (the non-substrates in the drug discovery data), whereas "negative" refers to compounds which do not exhibit this pharmaceutical activity ("antitargets"). "True positives" are thus active compounds which have correctly been classified as active, and "false positives" are inactive compounds which have wrongly been classified as active. The accuracy of a classification process defined as the portion of true positives *and* true negatives in the population of all instances, $A = (TP + TN)/(TP + TN + FP + FN)$.

3.2 Feature subsets

For both data sets used we determined different feature subsets, i. e., out of the original data matrix D we computed a new matrix D'. For the FS subsets (IG and wrapper), D and D' differ only in the amount of columns (attributes), for the PCA subsets they differ in the number of columns *and* in their interpretation (cf. Section 2.2).

For extracting the wrapper subsets we used WEKA's *wrapper subset evaluator* in combination with the *best first* search method (i. e., searching the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility). A paired t-test is used to compute the probability if other subsets may perform substantially better. If this probability is lower than a pre-defined threshold the search is stopped. The result is a (heuristically) optimal feature subset for the applied learning algorithm. For this paper, neither a maximum nor a minimum number of features is pre-defined. The optimum number of features is automatically determined within the wrapper subset evaluator (between 4 and 18 features for our evaluations).

As a filter approach we ranked the attributes with respect to their information gain. As mentioned in Section 2.1, this ranking is independent of a specific learning algorithm and contains – before selecting a subset – all attributes.

For dimensionality reduction, we studied PCA. In the literature, several variants appear. We investigated the differences of three such variants (denoted by PCA_1, PCA_2 and PCA_3) in terms of the resulting classification accuracy. For all subsets based on PCA we first performed a mean shift of all features such that the mean for each feature becomes 0. We denote the resulting feature-instance matrix as M. Based on this first preprocessing step we define three variants of the PCA computation. The resulting PCA subsets contain min(features, instances) linear combinations of the original attributes (out of which a subset is selected).

PCA_1: The eigenvalues and eigenvectors of the covariance matrix of M (cf. Section 2.2) are computed. The new attribute values are then computed by multiplying M with the eigenvectors of Cov(M).

PCA_2: The eigenvalues and eigenvectors of the correlation matrix of M (cf. Section 2.2) are computed. The new attribute values are then computed by multiplying M with the eigenvectors of Corr(M).

PCA_3: Each feature of M is normalized by its standard deviation (i. e., z-scored). These normalized values are used for computing eigenvalues and eigenvectors (i. e., there is no difference between the covariance and the correlation coefficient) and also for the computation of the new attributes.

3.3 Machine Learning Methods

For evaluating the classification performance of the reduced feature sets we used six different machine learning methods. For detailed information about these methods, the reader is referred to the respective references given.

Experiments were performed with a support vector machine (SVM) based on the sequential minimal optimization algorithm using a polynomial kernel with an exponent of 1 (Platt, 1998); a k-nearest neighbors (kNN) classifier using different values of k (1 to 9) (Cover and Hart, 1995); a bagging ensemble learner using a pruned decision tree as base learner (Breiman, 1996); a single J.48 decision tree based on Quinlans C4.5 decision tree algorithm (Quinlan, 1993); a random forest (RandF) classifier using a forest of random trees (Breiman, 2004); and a Java implementation (JRip) of a propositional rule learner, called RIPPER (repeated incremental pruning to produce error reduction (Cohen, 1995)).

4. Experimental Results

For all feature sets except the wrapper subsets we measured the classification performance for subsets consisting of the n "best ranked" features (n varies between 1 and 100). For the information gain method, the top n information gain ranked original features were used. For the PCA subsets, the first n principle components capturing most of the variability of the original attributes were used.

The classification accuracy was determined using a 10-fold cross-validation. The results are shown separately for the two data sets. Only the kNN(1) results are shown since $k = 1$

yielded the best results. For comparison we also classified completely new data (separating feature reduction and classification process). In most cases the accuracy was similar.

4.1 E-Mail Data

Table 1 shows the *average* classification accuracy for the information gain subsets and the PCA subsets over all subsets of the top n features (for IG) and PCs (for PCA), respectively, ($n = 1, \ldots, 100$). Besides, the average classification accuracy over all algorithms is shown (AVG.). The best and the worst average results for each learning algorithm over the feature reduction methods are highlighted in bold and italic letters, respectively. The best overall result over all feature sets and all learning algorithms is marked with an asterisk.

Table 1: E-mail data – average overall classification accuracy (in %).

	SVM	kNN(1)	Bagging	J.48	RandF	JRip	AVG.
Infogain	*99.20*	*99.18*	*99.16*	*99.16*	*99.21*	*99.18*	*99.18*
PCA_1	99.26	**99.70**	**99.68**	**99.70**	* **99.77**	99.67	99.63
PCA_2	**99.55**	99.69	99.67	99.69	99.75	**99.68**	**99.67**
PCA_3	99.54	99.64	99.65	99.64	99.65	99.64	99.63

Table 2 shows the *best* classification accuracy for all feature subsets (including the wrapper subsets) and for a classification based on the complete feature set. Table 2 also contains the information how many original features (for FS) and how many principal components were needed to achieve the respective accuracy. Again, the best and worst results are highlighted.

Table 2: E-mail data – best overall classification accuracy (in %).

	SVM	kNN(1)	Bagging	J.48	RandF	JRip
All features	99.75 *230 attr.*	99.70 *230 attr.*	99.71 *230 attr.*	99.65 *230 attr.*	99.73 *230 attr.*	99.66 *230 attr.*
Wrapper *fixed set*	*99.61* *7 attr.*	*99.60* *5 attr.*	*99.61* *4 attr.*	*99.61* *7 attr.*	*99.67* *11 attr.*	*99.64* *7 attr.*
Infogain	**99.76** ***100 attr.***	99.71 *50 attr.*	99.70 *50 attr.*	99.71 *100 attr.*	99.78 *80 attr.*	99.72 *90 attr.*
PCA_1	*99.65* *90 PCs*	99.74 *40 PCs*	99.69 *40 PCs*	99.75 *30 PCs*	* **99.82** ***70 PCs***	99.73 *5 PCs*
PCA_2	99.67 *90 PCs*	**99.75** ***40 PCs***	**99.72** ***30 PCs***	**99.78** ***60 PCs***	99.80 *15 PCs*	**99.76** ***40 PCs***
PCA_3	99.65 *100 PCs*	99.73 *5 PCs*	99.71 *20 PCs*	99.71 *4 PCs*	99.79 *15 PCs*	99.73 *50 PCs*

Feature subset selection. A comparison of the two FS methods shows that the best accuracy achieved with IG subsets are better than the wrapper results (see Table 2). Nevertheless, when looking at the size of the subsets with the best accuracy it can be seen that the wrapper subsets are very small. Figure 1 shows the degradation in the classification accuracy when the number of features in the IG subsets is reduced. Interestingly, all machine learning methods show the same behavior without any significant differences. The accuracy is quite stable until the subsets are reduced to 30 or less features, then the overall classification accuracy tends to decrease proportionally to the reduction of features.

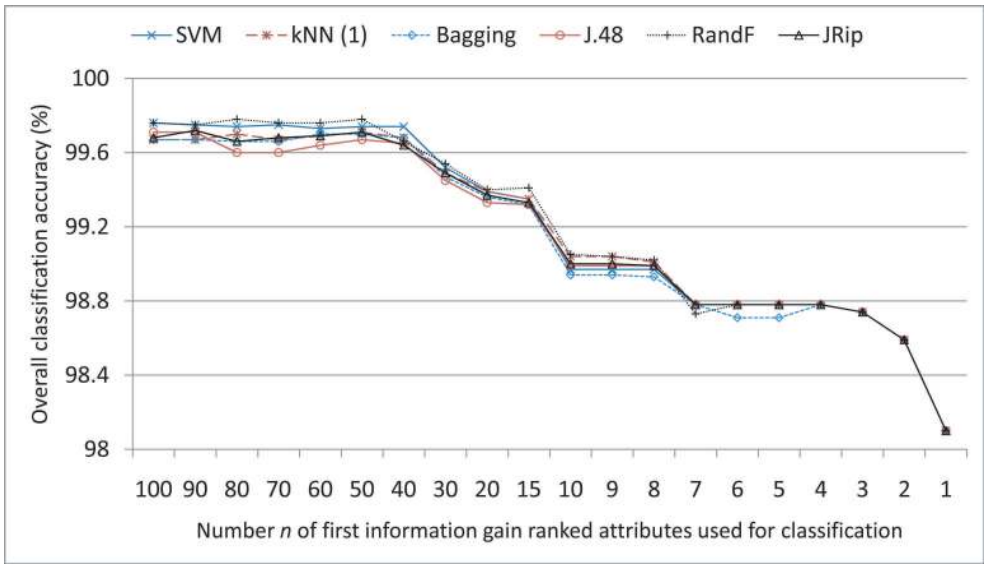

Figure 1: E-mail data – information gain subsets

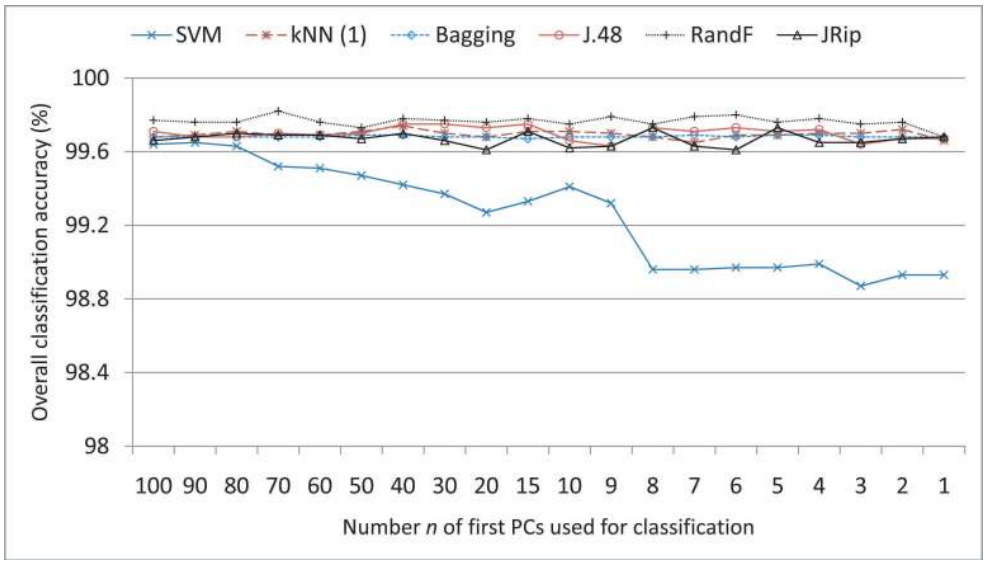

Figure 2: E-mail data – PCA_1 subsets

Comparing the wrapper results with the IG results with the same number of attributes (Figure 1 (4 to 11 features), and Table 2), it can be seen that the wrapper approach clearly outperforms IG. Moreover, the classification accuracy achieved with the wrapper subsets is only slightly worse than the accuracy achieved with the complete feature set, but with a reduction of about 96% of the feature space.

PCA. Figure 2 shows the overall classification accuracy for PCA_1. It is very stable regardless of the number of principle components used. Only the accuracy of the SVM method clearly decreases with a smaller number of PCs. The accuracy for the other two PCA

variants PCA_2 and PCA_3 is very similar (see Tables 1 and 2, due to space limitations the curves are not shown here).

Explaining the variance. Even though the classification results for all three PCA variants are similar, it is very interesting to note that when the correlation matrix is used instead of the covariance matrix to compute the eigenvectors and eigenvalues (as it is the case for PCA_2 and PCA_3) the fraction of the variance captured by the first n PCs (i. e., the accumulated percentage of the first n eigenvalues) decreases remarkably.

Table 3: E-mail data – % variance captured by first n PCs (max. dim: 230)

PCs	100	80	60	40	20	10	8	6	5	4	3	2	1
PCA_1	99.4%	98.9%	97.8%	95.3%	87.0%	75.7%	71.7%	66.0%	62.7%	58.9%	54.0%	48.0%	38.0%
$PCA_{2,3}$	67.7%	58.9%	49.4%	38.3%	24.7%	15.8%	13.7%	11.5%	10.3%	9.0%	7.7%	6.3%	3.9%

Algorithms. Although the overall classification accuracy is very good in general, when comparing the different machine learning methods is can be seen that random forest achieves the best results for all of the reduced feature sets used, and that the SVM classifier seems to be very sensitive to the size of the PC subsets. For the average PCA results (Table 1, Figure 2) SVM shows the lowest classification accuracy. When the complete feature set is used, the SVM results are slightly better than the random forest results (Table 2).

4.2 Drug Discovery Data

Tables 4 and 5 show average and best classification results for the drug discovery data.

Table 4: Drug discovery data – average overall classification accuracy (in %).

	SVM	kNN(1)	Bagging	J.48	RandF	JRip	AVG.
Infogain	**69.25**	**67.55**	70.63	**68.47**	68.04	**70.32**	**69.04**
PCA_1	63.50	*65.87*	*65.84*	*61.81*	*65.03*	*65.74*	*64.63*
PCA_2	*61.05*	66.39	69.27	65.27	67.16	65.92	65.84
PCA_3	68.78	67.28	*** 71.02**	63.76	**69.66**	67.06	67.93

Table 5: Drug discovery data – best overall classification accuracy (in %).

	SVM	kNN(1)	Bagging	J.48	RandF	JRip
All features	70.67 *367 attr.*	73.89 *367 attr.*	74.30 *367 attr.*	*64.24* *367 attr.*	73.52 *367 attr.*	69.09 *367 attr.*
Wrapper *fixed set*	**77.48** ***18 attr.***	**79.91** ***6 attr.***	**79.51** ***10 attr.***	**79.53** ***6 attr.***	*** 79.93** ***6 attr.***	**79.89** ***6 attr.***
Infogain	72.70 *60 attr.*	73.08 *80 attr.*	74.31 *20 attr.*	71.11 *1 attr.*	71.89 *7 attr.*	73.52 *2 attr.*
PCA_1	70.69 *60 PCs*	*73.07* *15 PCs*	*68.68* *15 PCs*	65.87 *15 PCs*	*69.48* *4 PCs*	69.09 *6 PCs*
PCA_2	*65.89* *60 PCs*	71.89 *60 PCs*	73.08 *15 PCs*	68.29 *60 PCs*	73.89 *40 PCs*	*68.69* *4 PCs*
PCA_3	73.89 *6 PCs*	73.09 *10 PCs*	75.90 *6 PCs*	69.09 *5 PCs*	76.69 *10 PCs*	71.48 *7 PCs*

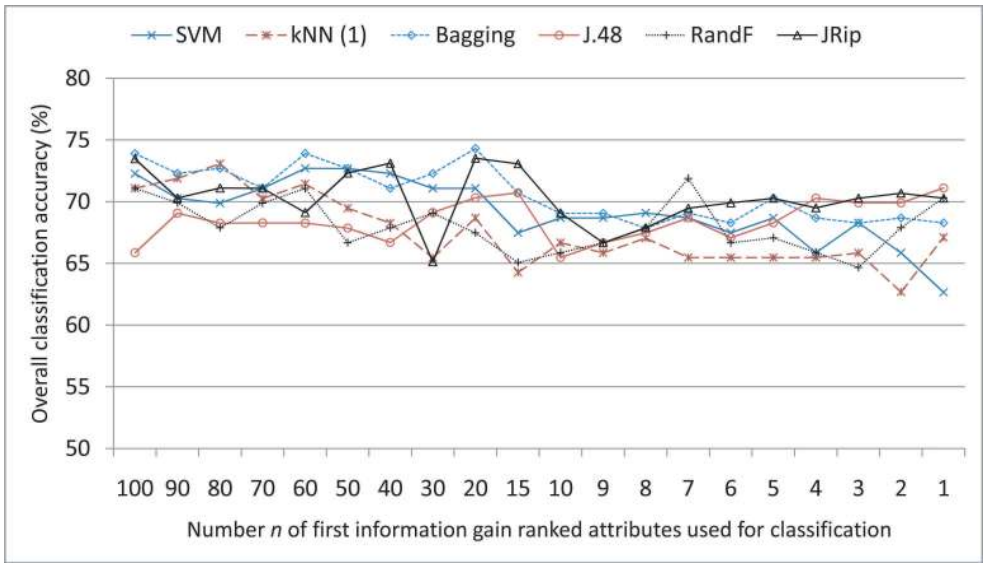

Figure 3: Drug discovery data – information gain subsets

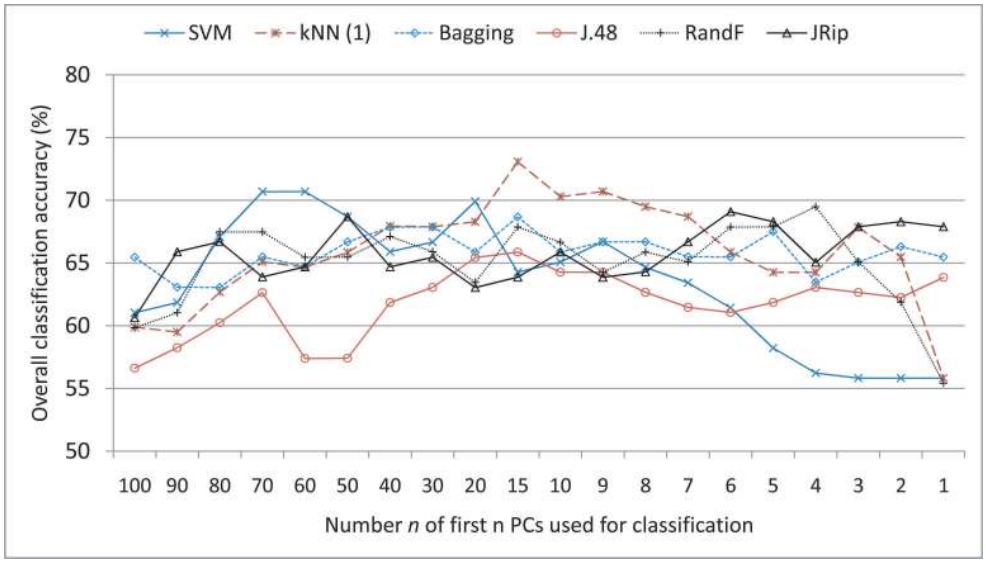

Figure 4: Drug discovery data – PCA_1 subsets

Feature subset selection. A very interesting observation is that the (again very small) wrapper subsets clearly outperform the complete feature set using all attributes and the IG subsets, in contrast to the results for the e-mail data also for larger IG subsets with 30-100 features (see Table 5). The three best wrapper results (kNN, random forest and JRip) were all achieved with a feature set containing only 6 attributes. Interestingly, only two of these attributes appear in all three subsets, the other attributes differ.

Figure 3 shows the classification performance for the IG subsets with different sizes. There is a clear difference between this curve and the curve shown in Figure 1 (IG subsets for the e-mail data). There is only a small decline in the accuracy when the number of attributes

is decreased, and even with very small IG gain subsets the classification performance remains acceptable (compared to large IG subsets).

PCA. Figure 4 shows the overall classification accuracy for PCA_1. The results are again very different from the results with the e-mail data set. Surprisingly, the accuracy between the first 90 to 100 PCs, as well as the results using only very few PC are much lower than the results for other PC numbers. When comparing PCA_1 and PCA_2 (Tables 4 and 5), it can be seen that the results for some classifiers change significantly. For example, the best classification accuracy for SVM decreases by 5% even though both subsets achieved the best result with the same number of PCs (60, see Table 5). On the other hand, for bagging and RF the accuracy improved by about 4%. The best bagging results were again achieved with the same number of PCs (15). The best PCA variant for this data set is PCA_3.

Explaining the variance. The percentage of variance captured by the first n principal components is much higher than for the e-mail data set (cf. Table 6). The reasons for this are the different sizes and the different characteristics of the data sets (see Section 3.1.

Table 6: Drug discovery data – % of variance captured by first n PCs (max. dim: 249).

PCs	100	80	60	40	20	10	8	6	5	4	3	2	1
PCA_1	100%	99.9%	99.9%	99.9%	99.9%	99.8%	99.6%	99.3%	98.7%	98.1%	97.0%	94.6%	87.9%
$PCA_{2,3}$	99.6%	99.1%	97.9%	95.2%	88.7%	79.9%	76.5%	71.5%	68.2%	63.6%	58.1%	51.4%	40.6%

Algorithms. Compared to the e-mail data set, the overall classification accuracy is much lower for all machine learning methods used. Comparing the six different algorithms, it can be seen that bagging achieves the best accuracy on the average results (Table 4). The best results for a given feature subset (Table 5) are either achieved with kNN, bagging or RF.

4.3 Runtimes

Figures 5 and 6 show the runtimes for the classification process (training *and* testing) for the e-mail and the drug discovery data set, respectively, for a ten-fold cross validation using information gain subsets with different numbers of features and for the complete feature set. Obviously, the time needed for the classification process decreases with fewer attributes. For all machine learning algorithms except kNN, a big amount of the classification time is spent in the training step. Once the model has been built, the classification (testing) of new instances can be performed rather quickly. As kNN does not build a model but computes all distances in the testing step, there is no training step for this approach. Comparing the classification times, it can be seen that kNN is the slowest classifier on the e-mail data set but the fastest on the drug discovery data set. The fastest classifier on the e-mail data set is the support vector classifier, the slowest classifier on the drug discovery data set is the bagging ensemble method.

Feature reduction runtimes. Table 7 shows the runtimes for different feature reduction processes, performed with WEKA on the complete data sets. It is very interesting to note the big gap between the time needed to compute the information gain ranking and the various wrapper methods. On the smaller drug discovery data set, the slowest wrapper (WrapRF, random forest) needed more than 48 minutes, on the e-mail data set more than

12 hours! On the e-mail data set, the slowest wrapper (kNN) needed even more than 20 hours, but was the fastest classifier on the drug discovery data set. For a fair comparison, we used WEKA's PCA routine (which is not the fastest available routine) for computing the PCs. It is slower than the computation the information gain ranking, but much faster than the wrapper subset selection methods.

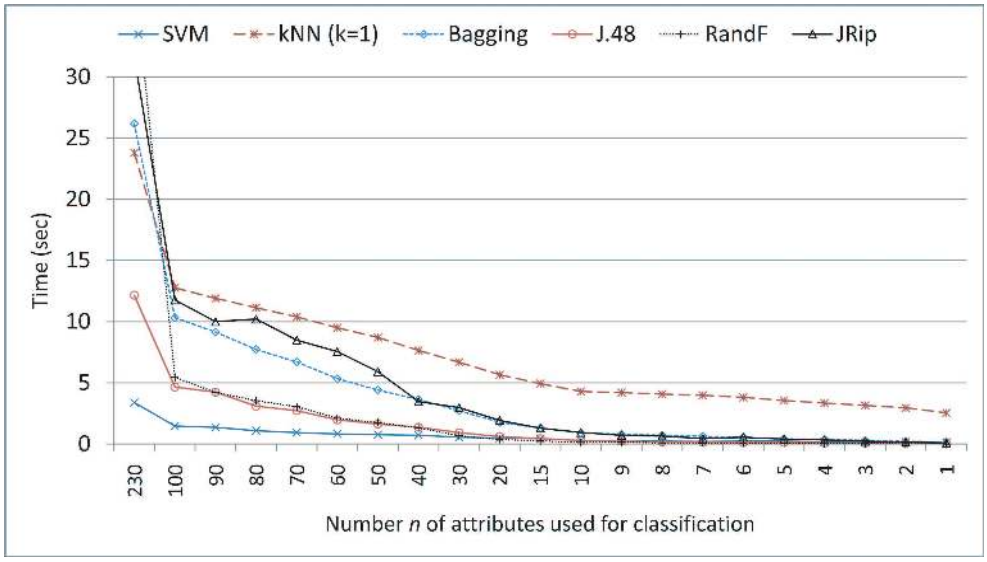

Figure 5: E-mail data – classification runtimes

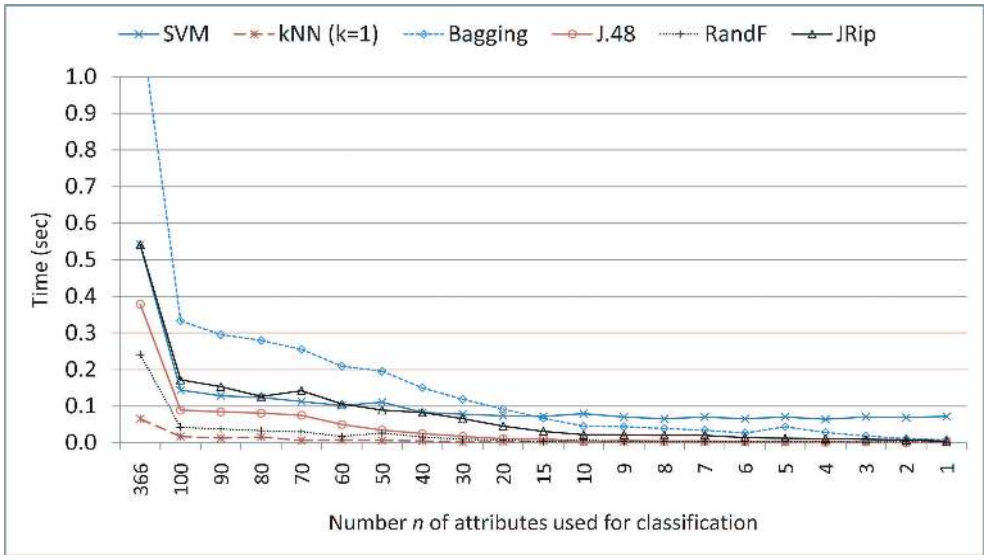

Figure 6: Drug discovery data – classification runtimes

Table 7: Subset selection runtimes (in seconds)

	IG	PCA	WrapSVM	WrapJ.48	WrapBag	WrapJRip	WrapRandF	WrapkNN
E-mail	5.12 e+0	8.10 e+1	3.06 e+3	4.44 e+3	7.20 e+3	1.26 e+4	4.46 e+4	7.42 e+4
DD	2.00 e−1	3.00 e+0	1.88 e+3	1.37 e+3	2.20 e+3	8.10 e+2	2.91 e+3	4.14 e+2

5. Conclusions

We have investigated the relationship between various feature reduction methods (feature subset selection as well as dimensionality reduction) and the resulting classification performance. More specifically, feature subsets determined with a wrapper method and information gain were compared to sets of linear combinations of the original features, computed with three variants of the principle component analysis. Extensive experiments performed with data sets from two very different application contexts, e-mail classification and drug discovery, lead to the following conclusions.

When looking specifically at the two data sets investigated in this paper, we note that the classification accuracy achieved with different feature reduction strategies is highly sensitive to the type of data. For the e-mail data with quite well established feature sets there is much less variation in the classification accuracy than for the drug discovery data. Wrapper methods clearly outperform IG on the drug discovery data, and also show acceptable classification accuracy on the e-mail data set. Although for the e-mail data the best overall IG subset achieves better results than the wrapper subsets, the wrapper subsets lead to better accuracy than IG subsets with comparable sizes. Among the machine learning algorithms investigated, the SVM accuracy was surprisingly low on the PCA subsets. Even though SVMs perform very well when all features or subsets of the original features are used, they achieve only the lowest accuracy for all three PCA subsets of the e-mail data. On the drug discovery data, SVMs achieve a reasonable accuracy only with a PCA_3 subset. The accuracy of most classifiers tends to be much less sensitive to the number of features when principal components are used instead of subsets of the original features, especially so for the e-mail data. This is not surprising, since the principal components in general contain information from *all* original features. However, it is interesting to note that on the e-mail data the SVM classifier is an exception: Its accuracy decreases clearly when fewer principal components are used, similar to the situation when feature subsets are used.

More generally speaking, the experimental results underline the importance of a feature reduction process. In many cases, in particular in application contexts where the search for the best feature set is is still an active research topic (such as in the drug discovery application discussed), the classification accuracy achieved with reduced feature sets is often significantly *better* than with the full feature set. In application contexts where feature sets are already well established the differences between different feature reduction strategies are much smaller.

Among the feature selection methods, wrappers tend to produce the smallest feature subsets with very competitive classification accuracy (in many cases the best over all feature reduction methods). However, wrappers tend to be much more expensive computationally than the other feature reduction methods. For dimensionality reduction based on PCA, it is important to note that the three variants considered in this paper tend to differ significantly

in the percentage of the variance captured by a fixed number of principal components, in the resulting classification accuracy, and particularly also in the number of principal components needed for achieving a certain accuracy. It has also been illustrated clearly that the percentage of the total variability of the data captured in the principal components used is *not necessarily* correlated with the resulting classification accuracy.

The strong influence of different feature reduction methods on the classification accuracy observed underlines the need for more investigation in the complex interaction between feature reduction and classification.

Acknowledgments. We gratefully acknowledge financial support from the CPAMMS-Project (grant# FS397001) in the Research Focus Area "Computational Science" of the University of Vienna, the Austrian Science Fund (grant# L344-N17), and the Austrian Research Promotion Agency (grant# B1-812074).

References

Apache Software Foundation. SpamAssassin open-source spam filter, 2006. `http://spamassassin.apache.org/`.

Avrim L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2004.

William W. Cohen. Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

Gordon V. Cormack and Thomas R. Lynam. TREC 2005 spam public corpora, 2005. `http://plg.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/foo`.

Thomas M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 8(6):373–389, 1995.

Gerhard F. Ecker. In silico screening of promiscuous targets and antitargets. *Chemistry Today*, 23:39–42, 2005.

Terrence Furey, Nello Cristianini, Nigel Duffyy, David W. Bednarski, and David Hauessler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16:906–914, 2000.

Wilfried N. Gansterer, Andreas G. K. Janecek, and Robert Neumayer. Spam filtering based on latent semantic indexing. In Micheal W. Berry and Malu Castellanos, editors, *Survey of Text Mining II - Clustering, Classification, and Retrieval*, volume 2, pages 165–185. Springer, 2008.

Paul Geladi and Bruce R. Kowalski. Partial least-squares regression: A tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.

Richard L. Gorsuch. *Factor Analysis*. Lawrence Erlbaum, 2nd edition, 1983.

Isabelle Guyon and Andre Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

Tom Howley, Michael G. Madden, Marie-Louise O'Connell, and Alan G. Ryder. The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data. *Knowledge Based Systems*, 19(5):363–370, 2006.

Ian T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

Daphne Koller and Mehran Sahami. Toward optimal feature selection. pages 284–292. Morgan Kaufmann, 1996.

Huan Liu and Hiroshi Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, 1998.

Koji Miyahara and Michael J. Pazzani. Collaborative filtering with the simple Bayesian classifier. *Pacific Rim International Conference on Artificial Intelligence*, pages 679–689, 2000.

Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. Feature selection algorithms: A survey and experimental evaluation. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, pages 306–313, Washington, DC, USA, 2002. IEEE Computer Society.

John C. Platt. Machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

Lubomir Popelinsky. Combining the principal components method with different learning algorithms. In *Proceedings of the ECML/PKDD2001 IDDM Workshop*, 2001.

Warren Buckler Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience, 1st edition, 2007.

Ross J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison Wesley, 1st edition, 2005.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2 edition, 2005.

B. Zdrazil, D. Kaiser, S. Kopp, P. Chiba, and G.F. Ecker. Similarity-based descriptors (SIBAR) as tool for qsar studies on p-glycoprotein inhibitors: Influence of the reference set. *QSAR and Combinatorial Science*, 26(5):669–678, 2007.