The Impact of Feature Extraction on the Performance of a Classifier: kNN, Naïve Bayes and C4.5

Mykola Pechenizkiy

Dept. of Computer Science and Information Systems, University of Jyväskylä, Jyväskylä, Finland
mpechen@cs.jyu.fi

Abstract. "The curse of dimensionality" is pertinent to many learning algorithms, and it denotes the drastic raise of computational complexity and the classification error in high dimensions. In this paper, different feature extraction techniques as means of (1) dimensionality reduction, and (2) constructive induction are analyzed with respect to the performance of a classifier. Three commonly used classifiers are taken for the analysis: kNN, Naïve Bayes and C4.5 decision tree. One of the main goals of this paper is to show the importance of the use of class information in feature extraction for classification and (in)appropriateness of random projection or conventional PCA to feature extraction for classification for some data sets. Two eigenvector-based approaches that take into account the class information are analyzed. The first approach is parametric and optimizes the ratio of between-class variance to the within-class variance of the transformed data. The second approach is a nonparametric modification of the first one based on the local calculation of the between-class covariance matrix. In experiments on benchmark data sets these two approaches are compared with each other, with conventional PCA, with random projection and with plain classification without feature extraction for each classifier.

1 Introduction

Knowledge discovery in databases (KDD) is a combination of data warehousing, decision support, and data mining that indicates an innovative approach to information management. KDD is an emerging area that considers the process of finding previously unknown and potentially interesting patterns and relations in large databases [8]. Current electronic data repositories are growing quickly and contain huge amount of data from commercial, scientific, and other domain areas. The capabilities for collecting and storing all kinds of data totally exceed the abilities to analyze, summarize, and extract knowledge from this data. Numerous data mining techniques have recently been developed to extract knowledge from these large databases. Fayyad in [8] introduced KDD as "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". The process comprises several steps, which involve data selection, data pre-processing, data transformation, application of machine learning techniques, and the interpretation and evaluation of patterns.

B. Kégl and G. Lapalme (Eds.): AI 2005, LNAI 3501, pp. 268 – 279, 2005.

In this paper we analyze the problems related to data transformation, before applying certain machine learning techniques. In Section 2 the data transformation approaches are seen from two different perspectives. The first one is related to the so-called "curse of dimensionality" problem [5] and the necessity of dimensionality reduction [2]. The second perspective comes from the assumption that in many data sets to be processed some individual features, being irrelevant or indirectly relevant for the purpose of analysis, form poor problem representation space. Corresponding ideas of constructive induction that assume the improvement of problem representation before application of any learning technique are presented.

Feature extraction (FE) for classification is aimed at finding such a transformation of the original space in order to produce new features, which would preserve class separability as much as possible and to form a new lower-dimensional problem representation space. Thus, FE accounts for both the perspectives, and, therefore, we believe that FE, when applied either on data sets with high dimensionality or on data sets including indirectly relevant features, can improve the performance of a classifier.

We consider different types of feature extraction techniques for classification in Section 3, including Principal Component Analysis (PCA), Random Projection (RP) and two class-conditional approaches to FE. We conduct a number of experiments on 20 UCI datasets, analyzing the impact of these FE techniques on the classification performance of the nearest neighbour classification, Naïve Bayes, and C4.5 decision tree learning. The results of these experiments are reported in Section 4. And then, in Section 5 we briefly summarize with the main conclusions and further research directions.

2 Poor Representation Spaces: "The Curse of Dimensionality" and Indirectly Relevant Features

In this section the two main reasons are presented why data transformation might be an important step to be undertaken before a certain machine learning technique is applied. The first issue is related to the so-called "curse of dimensionality" and the necessity for dimensionality reduction. The second issue is related to the potentially poor representation of the problem in terms of some irrelevant or indirectly relevant features that represent the data and the corresponding necessity to improve the representation.

2.1 Dimensionality Reduction

In many real-world applications, numerous features are used in an attempt to ensure accurate classification. If all those features are used to build up classifiers, then they operate in high dimensions, and the learning process becomes computationally and analytically complicated, resulting often in the drastic rise of classification error. Hence, there is a need to reduce the dimensionality of the feature space before classification. According to the adopted strategy dimensionality reduction techniques are divided into feature selection and feature transformation (also called feature

discovery). The key difference between feature selection and feature transformation is that during the first process a subset of original features only is selected while the second approach is based on the generation of completely new features [15]. Feature extraction is a dimensionality reduction technique that extracts a subset of new features from the original set of features by means of some functional mapping keeping as much information in the data as possible [10].

The essential drawback of all the methods that just assign weights to individual features is their insensitivity to interacting or correlated features. Also, in many cases some features are useful on one example set but useless or even misleading in another. That is why the transformation of the given representation before weighting the features in such cases can be preferable. However, feature extraction and subset selection are not, of course, totally independent processes and they can be considered as different ways of task representation. And the use of such techniques is determined by the purposes, and, moreover, sometimes feature extraction and selection methods are combined together in order to improve the solution.

2.2 Constructive Induction

Even, if the dimensionality of problem is relatively low, the problem is that most inductive learning approaches assume that the features used to represent instances are sufficiently relevant. However, it was shown experimentally that this assumption does not hold often for many learning problems. Some features may not be directly relevant, and some features may be redundant or irrelevant. Even those inductive learning approaches that apply feature selection techniques, and can eliminate irrelevant features and thus somehow account for the problem of high dimensionality, often fail to find good representation of data. This happens because of the fact that many features in their original representation are weakly or indirectly relevant to the problem. The existence of such features usually requires the generation of new, more relevant features that are some functions of the original ones. Such functions may vary from very simple as a product or a sum of a subset of the original features to very complex as a feature that reflects whether some geometrical primitive is present or absent in an instance. The discretization (quantization) of continuous features may serve for abstraction of some features when the reduction of the range of possible values is desirable. The original representation space can be improved for learning by removing less relevant features, adding more relevant features and abstracting features. We consider a constructive induction approach with respect to classification.

Constructive induction (CI) is a learning process that consists of two intertwined phases, one of which is responsible for the construction of the “best” representation space and the second concerns with generating hypotheses in the found space [16]. In Figure 1 we can see two problems – with a) high-quality, and b) low-quality representation spaces (RS). So, in a) points marked by “+” are easily separated from the points marked by “–” using a straight line or a rectangular border. But in b) “+” and “–” are highly intermixed that indicates the inadequateness of the original RS. A common approach is to search for complex boundaries to separate the classes. The constructive induction approach suggests searching for a better representation space where the groups are better separated, as in c).

However, in this paper the focus is on constructing new features from the original ones by means of some functional mapping that is known as feature extraction. We consider FE from both perspectives – as a constructive induction technique as a dimensionality reduction technique.

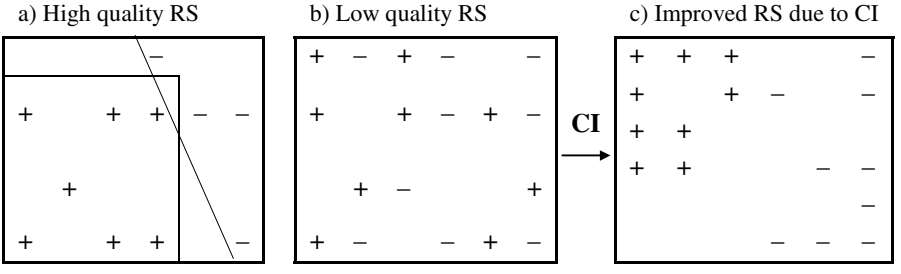

Fig. 1. High vs. low quality representation spaces (RS) for concept learning. Constructive induction (CI) aims to improve the quality of the low-quality RS [16]

3 Feature Extraction for Classification

Generally, feature extraction for classification can be seen as a search process among all possible transformations of the original feature set for the best one, which preserves class separability as much as possible in the space with the lowest possible dimensionality [10]. In other words we are interested in finding a projection **w**:

$$\mathbf{y} = \mathbf{w}^T \mathbf{x} \tag{1}$$

where **y** is a $k \times 1$ transformed data point (presented using k features), **w** is a $d \times k$ transformation matrix, and **x** is a $d \times 1$ original data point (presented using d features).

3.1 PCA

Principal Component Analysis (PCA) is a classical statistical method, which extracts a lower dimensional space by analyzing the covariance structure of multivariate statistical observations [12].

The main idea behind PCA is to determine the features that explain as much of the total variation in the data as possible with as few of these features as possible. The computation of the PCA transformation matrix is based on the eigenvalue decomposition of the covariance matrix **S** and therefore is computationally rather expensive.

$$\mathbf{w} \leftarrow eig\_decomposition\left(\mathbf{S} = \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T\right) \tag{2}$$

where n is the number of instances, $\mathbf{x}_i$ is the i-th instance, and **m** is the mean vector of the input data.

Computation of the principal components can be presented with the following algorithm:

1. Calculate the covariance matrix **S** from the input data.
2. Compute the eigenvalues and eigenvectors of **S** and sort them in a descending order with respect to the eigenvalues.
3. Form the actual transition matrix by taking the predefined number of components (eigenvectors).
4. Finally, multiply the original feature space with the obtained transition matrix, which yields a lower- dimensional representation.

The necessary cumulative percentage of variance explained by the principal axes is used commonly as a threshold, which defines the number of components to be chosen.

3.2 The Random Projection Approach

In many application areas like market basket analysis, text mining, image processing etc., dimensionality of data is so high that commonly used dimensionality reduction techniques like PCA are almost inapplicable because of extremely high computational time/cost.

Recent theoretical and experimental results on the use of random projection (RP) as a dimensionality reduction technique have attracted the DM community [6]. In RP a lower-dimensional projection is produced by means of transformation like in PCA but the transformation matrix is generated randomly (although often with certain constrains).

The theory behind RP is based on the Johnson and Lindenstrauss Theorem that says that any set of n points in a d-dimensional Euclidean space can be embedded into a k-dimensional Euclidean space – where k is logarithmic in n and independent of d – so that all pairwise distances are maintained within an arbitrarily small factor [1]. The basic idea is that the transformation matrix has to be orthogonal in order to protect data from significant distortions and try to preserve distances between the data points. Generally, orthogonalization of the transformation matrix is computationally expensive, however, Achlioptas showed a very easy way of defining (and also implementing and computing) the transformation matrix for RP [1]. So, according to [1] the transformation matrix **w** can be computed simply either as:

$$w_{ij} = \sqrt{3} \cdot \begin{cases} +1 & \text{with probability } 1/6 \\ 0 & \text{with probability } 2/3 \\ -1 & \text{with probability } 1/6 \end{cases}, \text{ or } w_{ij} = \begin{cases} +1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases} \quad (3)$$

RP as a dimensionality reduction technique was experimentally analyzed on image (noisy and noiseless) and text data (a newsgroup corpus) by Binghan and Mannila in [6]. Their results demonstrate that RP preserves the similarity of data vectors rather well (even when data is projected onto relatively small numbers of dimensions).

Fradkin and Madigan in [9] performed experiments (on 5 different data sets) with RP and PCA for inductive supervised learning. Their results show that although PCA predictively outperformed RP, RP is rather useful approach because of its computational advantages. Authors also indicated a trend in their results that the predictive performance of RP is improved with increasing the dimensionality when combining with the right learning algorithm. It was found that for those 5 data sets RP

is suited better for nearest neighbour methods, where preserving distance between data points is more important than preserving the informativeness of individual features, in contrast to the decision tree approaches where the importance of these factors is reverse. However, further experimentation was encouraged.

3.3 Class-Conditional Eigenvector-Based FE

In [17] it was shown that although PCA is the most popular feature extraction technique, it has a serious drawback, namely the conventional PCA gives high weights to features with higher variabilities irrespective of whether they are useful for classification or not. This may give rise to the situation where the chosen principal component corresponds to the attribute with the highest variability but without any discriminating power.

A usual approach to overcome the above problem is to use some class separability criterion [3], e.g. the criteria defined in Fisher's linear discriminant analysis and based on the family of functions of scatter matrices:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \tag{4}$$

where $\mathbf{S}_B$ in the parametric case is the between-class covariance matrix that shows the scatter of the expected vectors around the mixture mean, and $\mathbf{S}_W$ is the within-class covariance, that shows the scatter of samples around their respective class expected vectors.

A number of other criteria were proposed in [10]. Both parametric and nonparametric approaches optimize criterion (4) by using the *simultaneous diagonalization algorithm* [10].

It should be noticed that there is a fundamental problem with the parametric nature of the covariance matrices. The rank of $\mathbf{S}_B$ is at most the *number of classes-1*, and hence no more than this number of new features can be obtained.

The nonparametric method overcomes this problem by trying to increase the number of degrees of freedom in the between-class covariance matrix, measuring the between-class covariances on a local basis. The *k*-nearest neighbor (kNN) technique is used for this purpose.

A two-class nonparametric feature extraction method was considered in [10], and it is extended in [20] to the multiclass case. The algorithm for nonparametric feature extraction is the same as for parametric extraction. Simultaneous diagonalization is used as well, and the only difference is in calculating the between-class covariance matrix $\mathbf{S}_B$. In the nonparametric case the between-class covariance matrix is calculated as the scatter of the samples around the expected vectors of other classes' instances in the neighborhood.

A number of experimental studies where parametric and nonparametric class-conditional FE have been applied for kNN [20], dynamic integration of classifiers [19] and data with small sample size and high number of feature [11] were considered.

4 Experiments and Results

The experiments were conducted on 20 data sets with different characteristics taken from the UCI machine learning repository [7]. The main characteristics of the data sets are presented in the first four columns of Table 1, which includes the names of the data sets, the numbers of instances included in the data sets, the numbers of different classes of instances, and the numbers of different kinds of features (binarized categorical plus numerical) included in the instances. Each categorical feature was replaced with a redundant set of binary features, each corresponding to a value of the original feature.

In the experiments, the accuracy of 3-nearest neighbor classification (3*NN*), Naïve-Bayes (NB) learning algorithm, and C4.5 decision tree (C4.5) [18] was calculated. All they are well known in the data mining and machine learning communities and represent three different approaches to learning from data. The main motivation to use 3 different kinds of classifiers is that we expect different impact of FE on the representation space not only with respect to different data sets but also with respect to different classifiers. In particular, for kNN it is expected that FE can produce better neigbourhood, for C4.5 – better (more informative) individual features, and for Naïve Bayes – uncorrelated features.

For each data set 30 test runs of Monte-Carlo cross validation were made to evaluate classification accuracy with the four feature extraction approaches and without feature extraction. In each run, the data set is first split into the training set and the test set by stratified random sampling to keep class distributions approximately same. Each time 30 percent of the instances of the data set are first randomly picked up to the test set. The remaining 70 percent of instances form the training set, which is used for finding the feature-extraction transformation matrix **w**. The test environment was implemented within the WEKA framework (the machine learning library in Java) [21]. The classifiers from this library were used with their default settings.

For PCA we used a 0.85 variance threshold, and for RP we took the number of projected features equal to 75% of original space. We took all the features extracted by parametric FE as it was always equal to *number of classes-1*.

Main results are presented in the last three columns of Table 1. Each cell contains the ordered list of 5 symbols from A to E, which code different FE techniques. A is RP, B - PCA, C - PAR (parametric FE), D - NPAR (nonparametric FE), E - Plain (case when no FE technique has been applied). At the first position is symbol that corresponds to the highest accuracy and the last one – to the lowest accuracy. A hyphen, when used instead of comma between the symbols, denotes the fact that the difference between the corresponding accuracies is less than 1%.

It can be seen from the table that for some data sets FE has no effect or deteriorates the classification accuracy compared to plain case E. In particular, for 3*NN* such situation is on 9 data sets from 20: Breast, Diabetes, Glass, Heart, Iris, Led, Monk-3, Thyroid, and Tic. For NB such situation is on 6 data sets from 20: Diabetes, Heart, Iris, Lymph, Monk-3, and Zoo. And for C4.5 such situation is on 11 data sets from 20: Car, Glass, Heart, Ionosphere, Led, Led17, Monk-1, Monk-3, Vehicle, Voting,

and Zoo. It can be seen also that often different FE techniques are the best for different classifiers and for different data sets. Nevertheless, class-conditional FE approaches, especially the nonparametric approach are most often the best comparing to PCA or RP. On the other hand it is necessary to point out that the parametric FE was very often the worst, and for 3*NN* and C4.5 parametric FE was the worst technique more often than RP. Such results highlight the very unstable behavior of parametric FE.

Thus as it could be expected different FE techniques are often suited in different contexts not only for different data sets but also for different classifiers.

Table 1. – Datasets characteristics and relative accuracy results of 3*NN*, Naïve Bayes and C4.5 classifiers that were applied in different data spaces produced by corresponding FE techniques

Dataset	inst	class	feat	3*NN*	NB	C4.5
Balance	625	3	20	C,E,D,B,A	C,E,D,B,A	C,D,E,A,B
Breast	286	2	38	E,B,D,C,A	C,D,B-E,A	C,B,D-E,A
Car	1728	4	21	D,C,E,B,A	D-C,E,B,A	E,D,C,B,A
Diabetes	768	2	8	E,D,B,A,C	D-E,A,B,C	D,A-E,B,C
Glass	214	6	9	E,B-D,A,C	D,B,E,A,C	C-E,D-B-A
Heart	270	2	13	E,A,D,B,C	E,A,D,B,C	E,A,D,B,C
Ionosphere	351	2	33	D,B,E,A,C	D,B,A,E,C	A-B-D-E,C
Iris Plants	150	3	4	A-B-D-E,C	E-A,D-C,B	A,E,B-D,C
LED	300	10	7	E,B-C-D,A	B,C,D,E,A	B-C-D-E,A
LED17	300	10	24	C,E,B-D,A	C,E,D,B,A	E,C,D,B,A
Liver	345	2	6	D,E,B,A,C	D,B,C,E,A	B,E,C,D,A
Lymph	148	4	36	B,D-E,C,A	E,B,D,C,A	B,E,D,C,A
Monk-1	432	2	15	D,E,B,A,C	D,B,C-E,A	E,D,A,B,C
Monk-2	432	2	15	D,E,C,B,A	D,C,A-B,E	D,E,B,C,A
Monk-3	432	2	15	D-E,B,C,A	C-D-E,B,A	E,D,A-B,C
Thyroid	215	3	5	E,A-B,D,C	E,A,D-B,C	B,E,A,C-D
Tic	958	2	27	B-E,D,A,C	B,D,C,A-E	B,E,D,A,C
Vehicle	846	4	18	D,E,A,B,C	D,C,B,A-E	E,D,A,B,C
Voting	435	2	48	A,B-D-E,C	D,A-B-E,C	E,D,A-B,C
Zoo	101	7	16	C,D,B-E,A	D-E,C,A-B	E,B-A,D,C

Figure 2 summarizes ranking results of the FE techniques according to the classifiers performance on 20 UCI data sets. Each bar on the histograms shows how many times an FE technique was the 1st, the 2nd, the 3rd, the 4th, or the 5th among the 20 possible. The number of times certain techniques got 1st-5th place is not necessarily integer since there were draws between 2, 3, or 4 techniques. In such cases each technique gets the ½, 1/3 or 1/4 score correspondingly.

It can be seen from the figure that there are many common patterns in the behavior of techniques for 3 different classifiers, yet there are some differences too. So, according the ranking results RP behavior is very similar with every classifier, PCA works better for C4.5, parametric FE is suited better for NB. Nonparametric FE is also better suited for NB, it is also good with 3*NN*. However, it is less successful for C4.5.

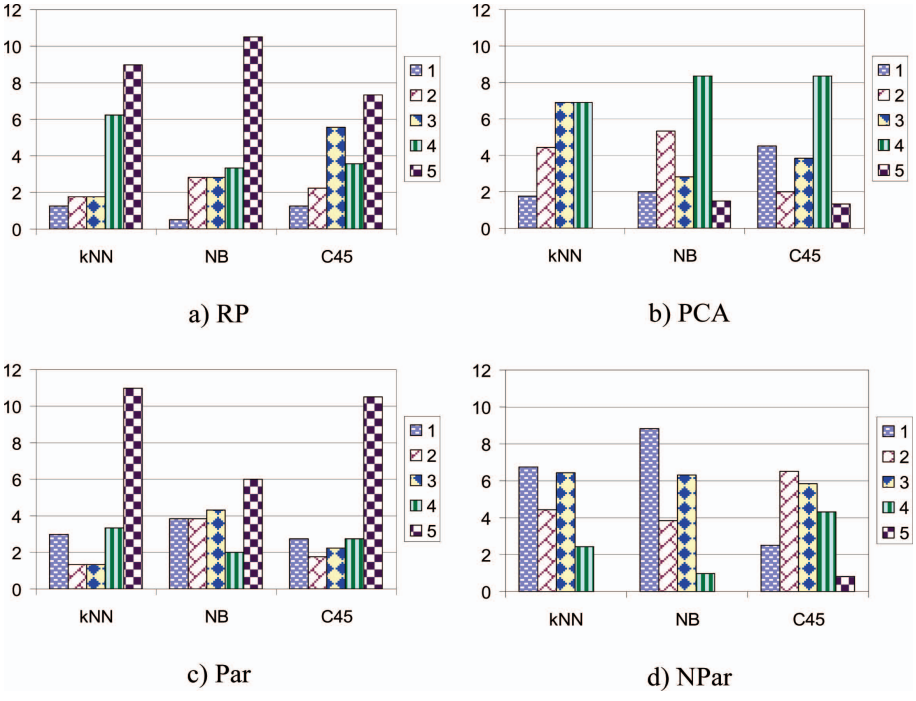

Fig. 2. Ranking of the FE techniques according to the results on 20 UCI data sets

In Table 2, averaged over 20 data sets, accuracy results are presented for each classifier. It can be seen from the table that among the FE techniques the nonparametric approach is always the best on average for each classifier, the second best is PCA, then parametric FE, and, finally, RP shows the worst results. Classification in the original space (*Plain*) was almost as good as in the space of extracted features produced by the nonparametric approach when kNN is used. However, when NB is used, *Plain* accuracy is significantly lower comparing to the situation when the nonparametric FE is applied. Still, this accuracy is as good as in situation when PCA is applied and significantly higher in situations when RP or the parametric FE is applied. For C4.5 the situation is also different. So, *Plain* classification is the best option on average. With respect to RP our results differ from the conclusions made in [9], where RP was found to be suited better for nearest neighbor methods and less satisfactory for decision trees (according to the results on 5 data sets). We can see from the two last columns of Table 2 that on average RP suits better for kNN than to C4.5 (*Plain-RP*) indeed (however the difference is only 0.5%). However, if we take into consideration PCA (*PCA-RP*), as in this context RP is often seen as an alternative for PCA, we can see that in fact RP produces new spaces that are better suited for C4.5 than to kNN (1.6%). It is interesting also to analyze these differences for NB. It can be seen that for *PCA-RP* the difference is the greatest while for *Plain-RP* the difference is the least. We believe that this is due to the poor performance of NB comparing to kNN and C4.5.

Table 2. Averaged over 20 data sets accuracy results

	RP	PCA	PAR	NPAR	Plain	PCA-RP(%)	Plain-RP(%)
kNN	.725	.774	.733	.808	.804	4.9	7.9
NB	.704	.767	.749	.799	.769	6.3	6.4
C4.5	.741	.775	.761	.806	.825	3.3	8.4

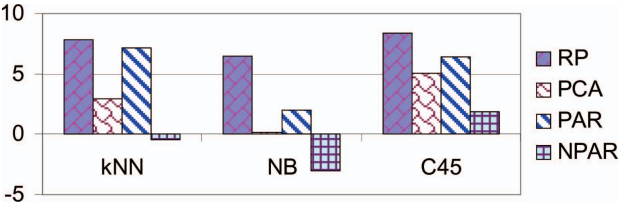

Fig. 3. The decrease/increase of accuracy due to the use of FE techniques, averaged over 20 data sets

In Figure 3 decrease/increase of accuracy due to the use of FE techniques is presented averaged over 20 data sets. It can be seen from the figure that on average, for 20 datasets analyzed FE has no effect on or deteriorates the classification accuracy. The only exception is combination of nonparametric FE and the NB classifier. In this situation averaged accuracy increases by 3%. In can be seen also that the nonparametric FE is the best among considered approaches from the accuracy perspective.

5 Conclusions

FE techniques are powerful tools that can significantly increase the classification accuracy producing better representation spaces or resolving the problem of "the curse of dimensionality". However, when applied blindly, FE may have no effect for the further classification or even deteriorate the classification accuracy. Moreover it is also possible that some data sets are so easy to learn that classification without any FE gains already the maximal possible accuracy and therefore it is hard to get any improvement due to FE.

In this paper, the experimental results show that for many data sets FE does increase classification accuracy.

We could see from the results that there is no best FE technique among the considered ones, and it is hard to say which one is the best for a certain classifier and/or for a certain problem, however according to the experimental results some major trends can be recognized.

Class-conditional approaches (and especially nonparametric approach) were often the best ones. This indicated the fact how important is to take into account class information and do not rely only on the distribution of variance in the data. At the same time it is important to notice that the parametric FE was very often the worst, and for 3*NN* and C4.5 the parametric FE was the worst more often than RP. Such results highlight the very unstable behavior of parametric FE.

One possibility to improve the parametric FE, we think, is to combine it with PCA or a feature selection approach in a way that a few principal components or the most useful for classification features are added to those extracted by the parametric approach.

Although it is logical to assume that RP should have more success in applications where the distances between the original data points are meaningful and/or for such learning algorithms that use distances between the data points, our results show that this is not necessarily true. However, data sets in our experiments have 48 features at most and RP is usually applied for problems with much higher dimensionality.

Time taken to build classification models with and without FE is not reported in this study, we also do not present here the analyses of number of features extracted by a certain FE technique. These important issues will be presented in our further study.

A volume of accumulated empirical (and theoretical) findings, some trends, and some dependencies with respect to data set characteristics and use of FE techniques have been discovered or can be discovered. In particular, it was shown that FE techniques are beneficial for data sets with highly correlated features [12]. The nonparametric FE was found to be successful especially for data sets with very limited sample sizes [11]. On the other hand, there are certain assumptions on the performance of classification algorithms under certain conditions.

Thus, potentially the adaptive selection of the most suitable data mining techniques for a data set at consideration (that is a really challenging problem) might be possible. We see our further research efforts in this direction.

We would like to emphasize also the possibility to conduct experiments on synthetically generated datasets that is beneficial from two perspectives. First, this allows generating, testing and validating hypothesis on data mining strategy selection with respect to a dataset at hand under controlled settings when some data characteristics are varied while the others are held unchangeable.

Acknowledgments. This research is partly supported by the COMAS Graduate School of the University of Jyväskylä, Finland. I would like to thank Dr. Alexey Tsymbal and Prof. Seppo Puuronen for their valuable comments and suggestions to the paper.

References

1. Achlioptas, D. Database-friendly random projections. Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Santa Barbara, California, ACM Press (2001)
2. Aivazyan, S.A. Applied statistics: classification and dimension reduction. Finance and Statistics, Moscow (1989)

3. Aladjem, M. Multiclass discriminant mappings. Signal Processing, Vol .35, (1994) 1-18
4. Aladjem, M. Parametric and nonparametric linear mappings of multidimensional data. Pattern Recognition, Vol.24(6) (1991) 543-553
5. Bellman, R., Adaptive Control Processes: A Guided Tour, Princeton University Press, (1961)
6. Bingham, E., Mannila, H. Random projection in dimensionality reduction: applications to image and text data. Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, San Francisco, California, (2001)
7. Blake, C.L., Merz, C.J. UCI Repository of Machine Learning Databases. Dept. of Information and Computer Science, University of California, Irvine CA, (1998)
8. Fayyad, U.M. Data Mining and Knowledge Discovery: Making Sense Out of Data, IEEE Expert, Vol. 11(5), (1996) 20-25
9. Fradkin, D., Madigan, D. Experiments with random projections for machine learning. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, Washington, D.C., (2003)
10. Fukunaga, K. Introduction to statistical pattern recognition. Academic Press, London (1990)
11. Jimenez, L., Landgrebe, D. High Dimensional Feature Reduction Via Projection Pursuit. PhD Thesis and School of Electrical & Computer Engineering Technical Report TR-ECE 96-5, (1995)
12. Jolliffe, I.T. Principal Component Analysis. Springer, New York, NY, (1986)
13. Kiang, M. A comparative assessment of classification methods, Decision Support Systems, Vol. 35, (2003) 441-454
14. Kohavi, R., Sommerfield, D., Dougherty, J. Data mining using MLC++: a machine learning library in C++. Tools with Artificial Intelligence, IEEE CS Press, (1996) 234-245
15. Liu, H. Feature Extraction, Construction and Selection: A Data Mining Perspective, ISBN 0-7923-8196-3, Kluwer Academic Publishers (1998)
16. Michalski, R.S.. Seeking Knowledge in the Deluge of Facts, Fundamenta Informaticae, Vol. 30, (1997) 283-297
17. Oza, N.C., Tumer, K. Dimensionality Reduction Through Classifier Ensembles. Technical Report NASA-ARC-IC-1999-124, Computational Sciences Division, NASA Ames Research Center, Moffett Field, CA, (1999)
18. Quinlan, J.R. C4.5 Programs for Machine Learning. San Mateo CA: Morgan Kaufmann, (1993)
19. Tsymbal A., Pechenizkiy M., Puuronen S., Patterson D.W. Dynamic integration of classifiers in the space of principal components, In: L.Kalinichenko, R.Manthey, B.Thalheim, U.Wloka (Eds.), Proc. Advances in Databases and Information Systems: 7th East-European Conf. ADBIS'03, Lecture Notes in Computer Science, Vol. 2798, Heidelberg: Springer-Verlag (2003) 278-292
20. Tsymbal A., Puuronen S., Pechenizkiy M., Baumgarten M., Patterson D. Eigenvector-based feature extraction for classification. In Proc. 15th Int. FLAIRS Conference on Artificial Intelligence, Pensacola, FL, USA, AAAI Press (2002) 354-358
21. Witten I. and Frank E. Data Mining: Practical machine learning tools with Java implementations, Morgan Kaufmann, San Francisco, (2000)