

Eigenvector-based Feature Extraction for Classification

Alexey Tsymbal^{1,3}, Seppo Puuronen¹, Mykola Pechenizkiy²,
Matthias Baumgarten³, David Patterson³

¹Department of Computer Science and Information Systems,
University of Jyväskylä, P.O.Box 35, FIN-40351, Jyväskylä, Finland
alexey@cs.jyu.fi

²Niilo Mäki Institute, Jyväskylä, Finland

³Northern Ireland Knowledge Engineering Laboratory, University of Ulster, U.K.

Abstract

This paper shows the importance of the use of class information in feature extraction for classification and inappropriateness of conventional PCA to feature extraction for classification. We consider two eigenvector-based approaches that take into account the class information. The first approach is parametric and optimizes the ratio of between-class variance to within-class variance of the transformed data. The second approach is a nonparametric modification of the first one based on local calculation of the between-class covariance matrix. We compare the two approaches with each other, with conventional PCA, and with plain nearest neighbor classification without feature extraction.

1. Introduction

Data mining is the process of finding previously unknown and potentially interesting patterns and relations in large databases. A typical data-mining task is to predict an unknown value of some attribute of a new instance when the values of the other attributes of the new instance are known and a collection of instances with known values of all the attributes is given.

In many applications, data, which is the subject of analysis and processing in data mining, is multidimensional, and presented by a number of features. The so-called “curse of dimensionality” pertinent to many learning algorithms, denotes the drastic raise of computational complexity and the classification error in high dimensions (Aha et al., 1991). Hence, the dimensionality of the feature space is often reduced before classification is undertaken.

Feature extraction (FE) is a dimensionality reduction technique that extracts a subset of new features from the original set by means of some functional mapping keeping as much information in the data as possible (Fukunaga 1990). Conventional Principal Component Analysis (PCA) is one of the most commonly used feature extraction techniques, that is based on extracting the axes on which the data shows the highest variability (Jolliffe 1986). Although this approach “spreads” out the data in the new

basis, and can be of great help in regression problems and unsupervised learning, there is no guarantee that the new axes are consistent with the discriminatory features in a classification problem. Unfortunately, this often is not taken into account by data mining researchers (Oza 1999). There are many variations on PCA that use local and/or non-linear processing to improve dimensionality reduction (Oza 1999), though they generally are also based solely on the inputs.

In this paper we consider two eigenvector-based approaches that use the within- and between-class covariance matrices and thus do take into account the class information. In the next section we consider conventional PCA and give a simple example of why PCA is not always appropriate to feature extraction for classification.

2. Conventional PCA

PCA transforms the original set of features into a smaller subset of linear combinations that account for most of variance of the original set (Jolliffe 1986).

The main idea of PCA is to determine the features, which explain as much of the total variation in the data as possible with as few of these features as possible. In PCA we are interested in finding a projection \mathbf{w} :

$$\mathbf{y} = \mathbf{w}^T \mathbf{x}, \quad (1)$$

where \mathbf{y} is a $p \times 1$ transformed data point, \mathbf{w} is a $p \times p'$ transformation matrix, and \mathbf{x} is a $p \times 1$ original data point. PCA can be done through eigenvalue decomposition of the covariance matrix \mathbf{S} of the original data:

$$\mathbf{S} = \sum_{i=1}^n (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T, \quad (2)$$

where n is the number of instances, \mathbf{x}_i is the i -th instance, and \mathbf{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 \mathbf{S} from the input data.
2. Compute the eigenvalues and eigenvectors of \mathbf{S} and sort them in a descending order with respect to 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 should be consulted in order to set a threshold, which defines the number of components to be chosen.

PCA has the following properties: (1) it maximizes the variance of the extracted features; (2) the extracted features are uncorrelated; (3) it finds the best linear approximation in the mean-square sense; and (4) it maximizes the information contained in the extracted features.

Although PCA has a number of advantages, there are some drawbacks. One of them is that PCA gives high weights to features with higher variabilities disregarding whether they are useful for classification or not. From Figure 1 one can see why it can be dangerous not to use the class information (Oza 1999). The first case shows the proper work of PCA where the first principal component corresponds to the variable with the highest discriminating power, but from the second case one can see that the chosen principal component is not always good for class discrimination.

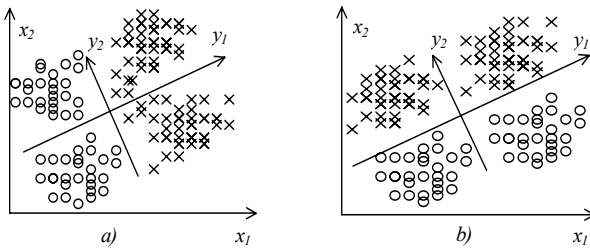


Fig. 1. PCA for classification: a) effective work of PCA, b) an irrelevant principal component was chosen wrt. to classification.

Nevertheless, conventional PCA is still often applied to feature extraction for classification by researchers.

3. Parametric Eigenvalue-based FE

Feature extraction for classification is a search among all possible transformations for the best one, which preserves class separability as much as possible in the space with the lowest possible dimensionality (Aladjem, 1994). The usual decision is to use some class separability criterion, based on a family of functions of scatter matrices: the within-class covariance, the between-class covariance, and the total covariance matrices.

The within-class covariance matrix shows the scatter of samples around their respective class expected vectors:

$$\mathbf{S}_W = \sum_{i=1}^c n_i \sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})(\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})^T, \quad (3)$$

where c is the number of classes, n_i is the number of instances in a class i , $\mathbf{x}_j^{(i)}$ is the j -th instance of i -th class, and $\mathbf{m}^{(i)}$ is the mean vector of the instances of i -th class.

The between-class covariance matrix shows the scatter of the expected vectors around the mixture mean:

$$\mathbf{S}_B = \sum_{i=1}^c n_i (\mathbf{m}^{(i)} - \mathbf{m})(\mathbf{m}^{(i)} - \mathbf{m})^T, \quad (4)$$

where c is the number of classes, n_i is the number of instances in a class i , $\mathbf{m}^{(i)}$ is the mean vector of the instances of i -th class, and \mathbf{m} is the mean vector of all the input data.

The total covariance matrix shows the scatter of all samples around the mixture mean. It can be shown analytically that this matrix is equal to the sum of the within-class and between-class covariance matrices (Fukunaga 1990):

$$\mathbf{S} = \mathbf{S}_B + \mathbf{S}_W. \quad (5)$$

One possible criterion based on the between- and within-class covariance matrices (3) and (4) to be optimized for feature extraction transformation (1) is defined in Fisher linear discriminant analysis:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}. \quad (6)$$

A number of other criteria were proposed in (Fukunaga 1990). The criterion (6) and some other relevant criteria may be optimized by the following algorithm often called *simultaneous diagonalization* (Fukunaga 1990):

1. Transformation of \mathbf{X} to \mathbf{Y} : $\mathbf{Y} = \mathbf{\Lambda}^{-1/2} \mathbf{\Phi}^T \mathbf{X}$, where $\mathbf{\Lambda}$ and $\mathbf{\Phi}$ are the eigenvalues and eigenvectors matrices of \mathbf{S}_W .
2. Computation of \mathbf{S}_B in the obtained \mathbf{Y} space.
3. Selection of m eigenvectors of \mathbf{S}_B , $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_m$, which correspond to the m largest eigenvalues.
4. Finally, new feature space $\mathbf{Z} = \boldsymbol{\Psi}_m^T \mathbf{Y}$, where $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_m]$, can be obtained.

It should be noted that there is a fundamental problem with the parametric nature of the covariance matrices. The features extracted with the parametric approach are suboptimal in the Bayes sense. The rank of the between-class covariance matrix (4) is at most $c-1$ (because it is the summation of c rank one matrices and only $c-1$ of them are independent), and hence no more than $c-1$ of the eigenvalues will be nonzero. The nonparametric method for feature extraction overcomes the above-mentioned problem.

4. Nonparametric Eigenvalue-based FE

The nonparametric method tries to increase the number of degrees of freedom in the between-class covariance matrix (4), measuring the between-class covariances on a local basis. K-nearest neighbor (kNN) technique is used for this purpose.

A two-class nonparametric feature extraction method was considered in (Fukunaga 1990), and it is extended in this paper to the multiclass case. The algorithm for

nonparametric feature extraction is the same as for the parametric extraction (Section 3). Simultaneous diagonalization is used as well, and the difference is only in calculation of the between-class covariance matrix. In the nonparametric between-class covariance matrix, the scatter of the samples around the expected vectors of other classes’ instances in the neighborhood is calculated:

$$\mathbf{S}_B = \sum_{i=1}^c n_i \sum_{k=1}^{n_i} w_{ik} \sum_{\substack{j=1 \\ j \neq i}}^c (\mathbf{x}_k^{(i)} - \mathbf{m}_{ik^*}^{(j)})(\mathbf{x}_k^{(i)} - \mathbf{m}_{ik^*}^{(j)})^T, \quad (7)$$

where $\mathbf{m}_{ik^*}^{(j)}$ is the mean vector of the nNN instances of j -th class, which are nearest neighbors to $\mathbf{x}_k^{(i)}$. The number of nearest instances nNN is a parameter, which should be set in advance. In (Fukunaga 1990) it was proposed to use nNN equal to 3, but without any justification. The coefficient w_{ik} is a weighting coefficient, which shows importance of each summand in (7). The goal of this coefficient is to assign more weight to those elements of the matrix, which involve instances lying near the class boundaries and thus more important for classification. We generalize the two-class version of this coefficient proposed in (Fukunaga 1990) to the multiclass case:

$$w_{ik} = \frac{\min_j \{d^\alpha(\mathbf{x}_k^{(i)}, \mathbf{x}_{nNN}^{(j)})\}}{\sum_{j=1}^c d^\alpha(\mathbf{x}_k^{(i)}, \mathbf{x}_{nNN}^{(j)})}, \quad (8)$$

where $d(\mathbf{x}_k^{(i)}, \mathbf{x}_{nNN}^{(j)})$ is the distance from $\mathbf{x}_k^{(i)}$ to its nNN -nearest neighbor of class j , and α is a parameter which should be set in advance. In (Fukunaga 1990) the parameter α equal to 1 was used, but without any justification.

In the next section we consider our experiments where we analyze and compare the described above feature-extraction techniques.

5. Experiments

The experiments were conducted on 21 data sets with different characteristics taken from the UCI machine learning repository (Blake et al., 1998). The main characteristics of the data sets are presented in 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 (categorical and numerical) included in the instances. The pre-selected values for the α and nNN are included in the table as well. In (Tsybaly et al., 2001) we have presented results of experiments with several feature selection techniques on these data sets.

In the experiments, the accuracy of 3-nearest neighbor classification based on the heterogeneous Euclidean-overlap metric was measured to test the feature extraction approaches. Categorical features were binarized as it was

done in the correlation-based feature selection experiments in (Hall et al., 2000). Each categorical feature was replaced with a redundant set of binary features, each corresponding to a value of the original feature.

Table 1. Characteristics of the data sets

| Data set | Instances | Classes | Features | | α | nNN |
|-------------|-----------|---------|-------------|-----------|----------|-------|
| | | | Categorical | Numerical | | |
| Balance | 625 | 3 | 0 | 4 | 1/3 | 255 |
| Breast | 286 | 2 | 9 | 0 | 5 | 1 |
| Car | 1728 | 4 | 6 | 0 | 5 | 63 |
| Diabetes | 768 | 2 | 0 | 8 | 1/5 | 127 |
| Glass | 214 | 6 | 0 | 9 | 1 | 1 |
| Heart | 270 | 2 | 0 | 13 | 1 | 31 |
| Ionosphere | 351 | 2 | 0 | 34 | 3 | 255 |
| Iris Plants | 150 | 3 | 0 | 4 | 1/5 | 31 |
| LED | 300 | 10 | 7 | 0 | 1/3 | 15 |
| LED17 | 300 | 10 | 24 | 0 | 5 | 15 |
| Liver | 345 | 2 | 0 | 6 | 3 | 7 |
| Lymph | 148 | 4 | 15 | 3 | 1 | 7 |
| MONK-1 | 432 | 2 | 6 | 0 | 1 | 1 |
| MONK-2 | 432 | 2 | 6 | 0 | 20 | 63 |
| MONK-3 | 432 | 2 | 6 | 0 | 1/3 | 1 |
| Soybean | 47 | 4 | 0 | 35 | 1 | 3 |
| Thyroid | 215 | 3 | 0 | 5 | 3 | 215 |
| Tic-Tac-Toe | 958 | 2 | 9 | 0 | 1 | 1 |
| Vehicle | 846 | 4 | 0 | 18 | 3 | 3 |
| Voting | 435 | 2 | 16 | 0 | 1/3 | 15 |
| Zoo | 101 | 7 | 16 | 0 | 1/20 | 7 |

For each data set 70 test runs of Monte-Carlo cross validation were made, first, to select the best α and nNN parameters, and after to evaluate the classification accuracy with the three feature extraction approaches and without any 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 instances of the data set are first randomly picked up to the test set. The remaining 70 percent instances form the training set, which is used for finding the feature-extraction transformation matrix (1). The test environment was implemented within the MLC++ framework (the machine learning library in C++) (Kohavi et al. 1996).

First, a series of experiments were conducted to select the best α and nNN coefficients for the nonparametric approach. The parameter α was selected from the set of 9 values: $\alpha \in \{1/20, 1/10, 1/5, 1/3, 1, 3, 5, 10, 20\}$, and the number of nearest neighbors nNN from the set of 8 values: $nNN = 2^i - 1, i = 1, \dots, 8, nNN \in \{1, 3, 7, 15, 31, 63, 127, 255\}$. The parameters were selected on the wrapper-like basis, optimizing the classification accuracy. For some data sets, e.g. LED and LED17, selection of the best parameters did not give almost any improvement in comparison with the considered in (Fukunaga 1990) $\alpha=1$ and $nNN=3$, and the classification accuracy varied within the range of one percent. It is necessary to note that the selection of the α and nNN parameters changed the ranking of the three feature extraction approaches from the accuracy point of view only on two data sets, thus demonstrating that the nonparametric approach is robust wrt. the built-in parameters. However, for some data sets the selection of the parameters had a significant positive effect on the

classification accuracy. For example, on the MONK-2 data set, accuracy is 0.796 when $\alpha=1$ and $nNN=3$, but it reaches 0.974 when $\alpha=20$ and $nNN=63$.

After, we have compared four classification techniques: the first three were based on the three considered above feature extraction approaches, and the last one did not use any feature extraction. For each feature selection technique, we have considered experiments with the best eigenvalue threshold of the following set $\{0.65, 0.75, 0.85, 0.9, 0.95, 0.97, 0.99, 0.995, 0.999, 1\}$.

The basic results of the experiments are presented in Table 2. First, average classification accuracies are given for the three extraction techniques: PCA, the parametric (Par) and nonparametric (NPar) feature extraction, and no feature extraction (Plain). The bold-faced and underlined accuracies represent the approaches that were significantly better than all the other approaches; the bold-faced only accuracies represent the approaches that were significantly worse on the corresponding data sets (according to the Student t-test with 0.95 level of significance). Then, the corresponding average numbers of extracted features are given. The remaining part contains the average extraction and the total expended time (in seconds) for the classification techniques. All the results are averaged over the 70 Monte-Carlo cross-validation runs.

Each row of Table 2 corresponds to one data set. The last two rows include the results averaged over all the data sets (the last row), and over the data sets containing

categorical features (the row before the last one).

From Table 2 one can see that the nonparametric approach has the best accuracy on average (0.824). Comparing the total average accuracy with the average accuracy on the categorical data sets, one can see that the nonparametric approach performs much better on the categorical data, improving the accuracy of the other approaches (as on the MONK data sets, and the Tic-Tac-Toe data set). The parametric approach is the second best. As we supposed, it is quite unstable, and not robust to different data sets' characteristics (as on the MONK-1,2 and Glass data sets). The case with no feature selection has the worst average accuracy.

The parametric approach extracts the least number of features on average (only 2.3), and it is the least time-consuming approach. The nonparametric approach is able to extract more features due to its nonparametric nature (9.9 on average), and still it is less time-consuming than the PCA and Plain classification.

Still, it is necessary to note that each feature extraction technique was significantly worse than all the other techniques at least on one data set (e.g., the Heart data set for the nonparametric approach), and it is a question for further research to define the dependencies between the characteristics of a data set and the type and parameters of the feature extraction approach best suited for it. For each data set, we have also pairwise compared each feature extraction technique with the others using the paired

Table 2. Results of the experiments

| Data set | Accuracy | | | | Features | | | | Extraction time, sec. | | | Total time, sec. | | | |
|---------------------|-------------|-------------|-------------|-------------|----------|-----|------|-------|-----------------------|------|------|------------------|------|-------|-------|
| | PCA | Par | NPar | Plain | PCA | Par | NPar | Plain | PCA | Par | Npar | PCA | Par | NPar | Plain |
| Balance | .827 | .893 | .863 | .834 | 4.0 | 1.0 | 2.0 | 4.0 | .00 | .09 | .21 | 3.11 | 1.02 | 1.87 | 2.55 |
| Breast | .721 | .676 | .676 | .724 | 16.5 | 1.0 | 33.7 | 51.0 | 2.66 | 3.10 | 4.00 | 5.33 | 3.31 | 9.32 | 5.88 |
| Car | .824 | .968 | .964 | .806 | 14.0 | 3.0 | 6.4 | 21.0 | .38 | .53 | .64 | 12.02 | 3.08 | 6.43 | 12.07 |
| Diabetes | .730 | .725 | .722 | .730 | 7.0 | 1.0 | 3.8 | 8.0 | .22 | .24 | .30 | 6.73 | 1.38 | 4.15 | 7.14 |
| Glass | .659 | .577 | .598 | .664 | 4.4 | 5.0 | 9.0 | 9.0 | .11 | .08 | .13 | .69 | .69 | 1.19 | 1.01 |
| Heart | .777 | .806 | .706 | .790 | 13.0 | 1.0 | 4.4 | 13.0 | .13 | .23 | .31 | 2.63 | .44 | 1.21 | 2.14 |
| Ionospher | .872 | .843 | .844 | .849 | 9.0 | 1.0 | 2.0 | 34.0 | 1.52 | 1.50 | 2.08 | 3.49 | 1.77 | 2.55 | 6.09 |
| Iris | .963 | .980 | .980 | .955 | 2.0 | 1.0 | 1.0 | 4.0 | .01 | .05 | .04 | .03 | .13 | .08 | .20 |
| LED | .646 | .630 | .635 | .667 | 7.0 | 7.0 | 7.0 | 14.0 | .13 | .39 | .49 | 1.61 | 1.92 | 1.99 | 2.17 |
| LED17 | .395 | .493 | .467 | .378 | 24.0 | 6.7 | 11.4 | 48.0 | 1.88 | 2.46 | 3.10 | 5.66 | 3.54 | 4.91 | 5.48 |
| Liver | .664 | .612 | .604 | .616 | 4.9 | 1.0 | 3.1 | 6.0 | .06 | .15 | .15 | 1.65 | .53 | 1.17 | 1.88 |
| Lymph | .813 | .832 | .827 | .814 | 31.4 | 3.0 | 32.0 | 47.0 | 1.58 | 2.04 | 2.50 | 3.39 | 2.23 | 4.39 | 1.96 |
| MONK-1 | .767 | .687 | .952 | .758 | 10.0 | 1.0 | 2.0 | 17.0 | .39 | .55 | .67 | 4.47 | 1.06 | 1.57 | 4.94 |
| MONK-2 | .717 | .654 | .962 | .504 | 8.0 | 1.0 | 2.0 | 17.0 | .40 | .60 | .70 | 3.76 | 1.08 | 1.60 | 4.96 |
| MONK-3 | .939 | .990 | .990 | .843 | 11.0 | 1.0 | 1.9 | 17.0 | .37 | .55 | .69 | 4.89 | 1.07 | 1.54 | 4.94 |
| Soybean | .992 | .987 | .986 | .995 | 7.8 | 1.0 | 2.2 | 35.0 | .17 | .45 | .44 | .23 | .46 | .47 | .07 |
| Thyroid | .921 | .942 | .933 | .938 | 4.0 | 2.0 | 2.0 | 5.0 | .05 | .03 | .05 | .52 | .35 | .33 | .69 |
| TicTacToe | .971 | .977 | .984 | .684 | 18.0 | 1.0 | 2.0 | 27.0 | .80 | .96 | 1.21 | 11.45 | 1.68 | 2.50 | 11.24 |
| Vehicle | .753 | .752 | .778 | .694 | 16.0 | 3.0 | 12.5 | 18.0 | .55 | .53 | .67 | 10.34 | 2.39 | 8.02 | 10.42 |
| Voting | .923 | .949 | .946 | .921 | 15.9 | 1.0 | 61.7 | 82.0 | 3.37 | 4.29 | 5.76 | 5.56 | 4.46 | 14.05 | 7.88 |
| Zoo | .937 | .885 | .888 | .932 | 15.1 | 6.4 | 6.5 | 36.0 | .62 | .85 | 1.09 | 1.03 | 1.00 | 1.28 | .78 |
| Average (categoric) | .787 | .795 | .845 | .730 | 15.5 | 2.9 | 15.1 | 34.3 | 1.14 | 1.48 | 1.90 | 5.38 | 2.22 | 4.51 | 5.66 |
| Average (total) | .801 | .803 | .824 | .766 | 11.6 | 2.3 | 9.9 | 24.4 | .73 | .94 | 1.20 | 4.22 | 1.60 | 3.36 | 4.50 |

Student *t*-test with 0.95 level of significance. Results of the comparison are given in Table 3. Columns 2-5 of the table contain results of the comparison of the technique corresponding to the row of the cell against the technique corresponding to the column using the paired *t*-test.

Each cell contains win/tie/loss information according to the *t*-test, and in parenthesis the same results are given for the eleven data sets including categorical features. For example, PCA has 8 wins against the parametric extraction on 21 data sets, and 5 of them are on categorical data sets.

Table 3. Results of the paired *t*-test (win/tie/loss information)

| | PCA | Parametric | Nonparametric | Plain |
|---------------|-------------------|-------------------|-------------------|-------------------|
| PCA | | 8/3/10 (5/0/6) | 8/1/13 (3/0/8) | 9/8/4 (5/5/1) |
| Parametric | 10/3/8 (6/0/5) | | 5/11/5 (2/6/3) | 11/5/5 (7/0/4) |
| Nonparametric | 13/1/8 (8/0/3) | 5/11/5 (3/6/2) | | 11/3/8 (8/0/3) |
| Plain | 4/8/9 (1/5/5) | 5/5/11 (4/0/7) | 8/3/11 (3/0/8) | |

From Tables 1, 2 one can see that classification without feature extraction is clearly the worst technique even for such data sets with relatively small numbers of features. This shows the so-called “curse of dimensionality” and necessity in feature extraction.

According to Table 3, among the three feature extraction techniques, the parametric and nonparametric techniques are the best on average, with the nonparametric technique being only slightly better than the parametric (3 wins versus 2 on the categorical data sets).

Conventional PCA was the worst feature extraction technique on average, which supports our expectations, as it does not take into account the class information. However, it was surprisingly stable. It was the best technique only on four data sets, but it was still the worst one only on three data sets (the best result).

On the categorical data sets the results are almost the same as on the rest of data sets. Only the nonparametric technique performs much better on the categorical data for this selection of the data sets, however, further experiments are necessary to check this finding.

6. Conclusions

PCA-based techniques are widely used for classification problems, though they generally do not take into account the class information and are based solely on inputs. Although this approach can be of great help in unsupervised learning, there is no guarantee that the new axes are consistent with the discriminatory features in a classification problem.

The experimental results supported our expectations. Classification without feature extraction was clearly the worst. This shows the so-called “curse of dimensionality” and necessity in feature extraction. Conventional PCA was the worst feature extraction technique on average and,

therefore, cannot be recommended for finding features that are useful for classification. The nonparametric technique was only slightly better than the parametric one on average. However, this can be explained by the selection of the data sets, which are relatively easy to learn and do not include significant nonnormal class distributions. Besides, better parameter tuning can be used to achieve better results with the nonparametric technique. This is an interesting topic for further research. The nonparametric technique performed much better on the categorical data for this selection of the data sets, however, further research is necessary to check this finding.

Another important topic for further research is to define the dependencies between the characteristics of a data set and the type and parameters of the feature extraction approach best suited for it.

Acknowledgements. This research is partly supported by the COMAS Graduate School of the University of Jyväskylä, Finland and NEURE project of Niilo Mäki Institute, Finland. We would like to thank the UCI ML repository of databases, domain theories and data generators for the data sets, and the MLC++ library for the source code used in this study.

References

- Aha, D., Kibler, D., Albert, M. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37-66.
- Aladjem, M. 1994. Multiclass discriminant mappings. *Signal Processing*, 35:1-18.
- Blake, C.L., Merz, C.J. 1998. UCI Repository of Machine Learning Databases [http:// www.ics.uci.edu/~mllearn/MLRepository.html]. Dept. of Information and Computer Science, University of California, Irvine CA.
- Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press, London.
- Hall, M.A. 2000. Correlation-based feature selection of discrete and numeric class machine learning. In *Proc. Int. Conf. On Machine Learning (ICML-2000)*, San Francisco, CA. Morgan Kaufmann, San Francisco, CA, 359-366.
- Jolliffe, I.T. 1986. *Principal Component Analysis*. Springer, New York, NY.
- Kohavi, R., Sommerfield, D., Dougherty, J. 1996. Data mining using MLC++: a machine learning library in C++. *Tools with Artificial Intelligence*, IEEE CS Press, 234-245.
- Oza, N.C., Tumer, K. 1999. Dimensionality Reduction Through Classifier Ensembles. Technical Report NASA-ARC-IC-1999-124, Computational Sciences Division, NASA Ames Research Center, Moffett Field, CA.
- Tsymbol A., Puuronen S., Skrypnik I., 2001. Ensemble feature selection with dynamic integration of classifiers, In: *Int. ICSC Congress on Computational Intelligence Methods and Applications CIMA'2001*, Bangor, Wales, U.K.