The Impact of Sample Reduction on PCA-based Feature Extraction for Supervised Learning

Mykola Pechenizkiy
Department of CS&ISs
Univ. of Jyväskylä, P.O. Box 35, Finland-40351
mpechen@cs.jyu.fi

Seppo Puuronen
Department of CS&ISs
Univ. of Jyväskylä, P.O. Box 35, Finland-40351
sepi@cs.jyu.fi

Alexey Tsymbal
Department of CS
Trinity College Dublin, Ireland
tsymbalo@cs.tcd.ie

ABSTRACT

"The curse of dimensionality" is pertinent to many learning algorithms, and it denotes the drastic raise of computational complexity and classification error in high dimensions. In this paper, different feature extraction (FE) techniques are analyzed as means of dimensionality reduction, and constructive induction with respect to the performance of Naïve Bayes classifier. When a data set contains a large number of instances, some sampling approach is applied to address the computational complexity of FE and classification processes. The main goal of this paper is to show the impact of sample reduction on the process of FE for supervised learning. In our study we analyzed the conventional PCA and two eigenvector-based approaches that take into account class information. The first class-conditional 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. The experiments are conducted on ten UCI data sets, using four different strategies to select samples: (1) random sampling, (2) stratified random sampling, (3) *kd*-tree based selective sampling, and (4) stratified sampling with *kd*-tree based selection. Our experiments show that if the sample size for FE model construction is small then it is important to take into account both class information and data distribution. Further, for supervised learning the nonparametric FE approach needs much less instances to produce a new representation space that result in the same or higher classification accuracy than the other FE approaches.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Management – *Database Applications, Data Mining*

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

Feature Extraction, Sample Reduction, Supervised Learning

SAC'06, April 23–27, 2006, Dijon, France.

1. INTRODUCTION

Numerous data mining techniques have recently been developed to extract knowledge from databases. Fayyad [7] introduced knowledge discovery from databases (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.

In this paper we analyze problems related to data transformation, phase before applying certain machine learning techniques. We consider feature extraction (FE) for supervised learning (SL). It is aimed at finding a transformation of the original space that would produce such new features, which preserve or improve class separability as much as possible and form a new lower-dimensional problem representation space (RS). Thus, FE for SL addresses (1) the so-called problem of "the curse of dimensionality" [3], which requires dimensionality reduction [1], and (2) the problem of poor RS caused by the presence of some irrelevant or indirectly relevant individual features. We consider these problems further in Section 2 and different types of FE techniques for SL in Section 3, including Principal Component Analysis (PCA), and two class-conditional approaches to FE.

When a data set contains a large number of instances, some sampling approach is often applied to address the computational complexity of FE and classification processes. The focus of this paper is within the study of sample reduction effect on three FE techniques with regard to the classification performance. In Section 4 we consider basic sampling strategies used in this study.

We conduct a number of experiments on ten UCI datasets, analyzing the impact of sample reduction on three FE techniques with regard to the classification performance of Naïve Bayes (NB). The results of these experiments are reported in Section 5. And then, in Section 6 we briefly summarize with the main conclusions and further research directions.

2. REPRESENTATION SPACE AND SL

In many real-world applications, numerous features are used in an attempt to better describe instances. 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. Different feature selection (FS) and FE techniques are used to cope with "the curse of dimensionality" and produce better RS. While FS is aimed at

selecting a subset of original features only, FE generates new features by means of some functional mapping keeping as much information in the data as possible [8].

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, FE 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 FE and selection methods are combined together in order to improve the solution.

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 [13]. 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 original RS can be improved for learning by removing less relevant features, adding more relevant features and abstracting features.

Constructive induction (CI) is a learning process that consists of two intertwined phases, one of which is responsible for the construction of the "best" RS and the second concerns with generating hypotheses in the found space [13]. In Figure 1 we can see two problems – with a) high-quality, and b) low-quality RSs. 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 which indicates the inadequateness of the original RS. A common approach is to search for complex boundaries to separate the classes. The CI approach suggests searching for a better representation space where the groups are better separated, as in c).

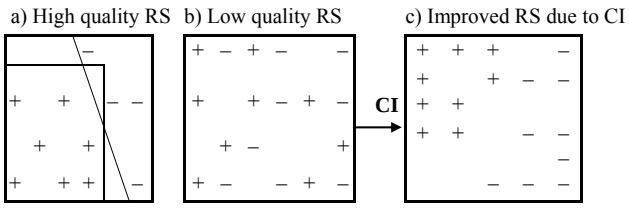

Figure 1. High vs. low quality RSs for concept learning. CI aims at improving the quality of the low-quality RS [13]

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 FE. We consider FE from both perspectives – as a constructive induction technique and as a dimensionality reduction technique.

3. FE FOR SUPERVISED LEARNING

Generally, FE for SL 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 [8]. In other words we are interested in finding a projection **w**:

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

where $\mathbf{y}$ is a $d \times 1$ transformed data point (presented using d features), $\mathbf{w}$ is a $k \times d$ transformation matrix, and $\mathbf{x}$ is a $k \times 1$ original data point (presented using k 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 [11].

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.

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

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 **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 Class-conditional Eigenvector-based FE

In [14] it was shown that although PCA is the most popular FE 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 [2], e.g. the criteria defined in Fisher's linear discriminant analysis and based on the family of functions of scatter matrices:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \quad (3)$$

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 [8]. Both parametric and nonparametric approaches optimize criterion (4) by using the *simultaneous diagonalization algorithm* [8].

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 (*k*NN) technique is used for this purpose. The algorithm for nonparametric FE 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, NB, and C4.5 [15], dynamic integration of classifiers [16] and data with small sample size and high number of feature [10] were considered.

4. RANDOM, STRATIFIED AND *KD*-TREE BASED SAMPLING

When a data set contains a large number of instances, some sampling strategy is normally applied before the FE and classification processes to reduce their computational time and cost.

In our study we apply random sampling (area in Figure 2 marked with dashed box), stratified random sampling (whole Figure 2), *kd*-tree based selective sampling (area in Figure 3 marked with dashed box), and stratified sampling with *kd*-tree based selection (whole Figure 3).

Random sampling and stratified random sampling are the most commonly applied strategies as they are straightforward and fast. In random sampling information about the distribution of instances by classes is disregarded. Therefore, intuitively, stratified sampling, which randomly selects instances from each group of instances (related to the corresponding class) separately, is preferable.

However, the assumption that instances are not uniformly distributed and some instances are more representative than others [12] motivates to apply a selective sampling approach. The main idea of selective sampling is to identify and select representative instances, so that fewer instances are needed to achieve similar (or even better) performance. The common approach to selective sampling is data partitioning (or data indexing) that is aimed to find some structure in data and then to select instances from each partition of the structure. Although there exist many data partitioning techniques (see e.g. [9] for an overview), we choose *kd*-tree for our study because of its simplicity, wide use and last but not least availability in WEKA library [17].

A *kd*-tree is a generalization of the simple binary tree which uses *k* features instead of a single feature to split instances in a multi-dimensional space [9]. The splitting is done recursively in each of the successor nodes until the node contains no more than a prede-

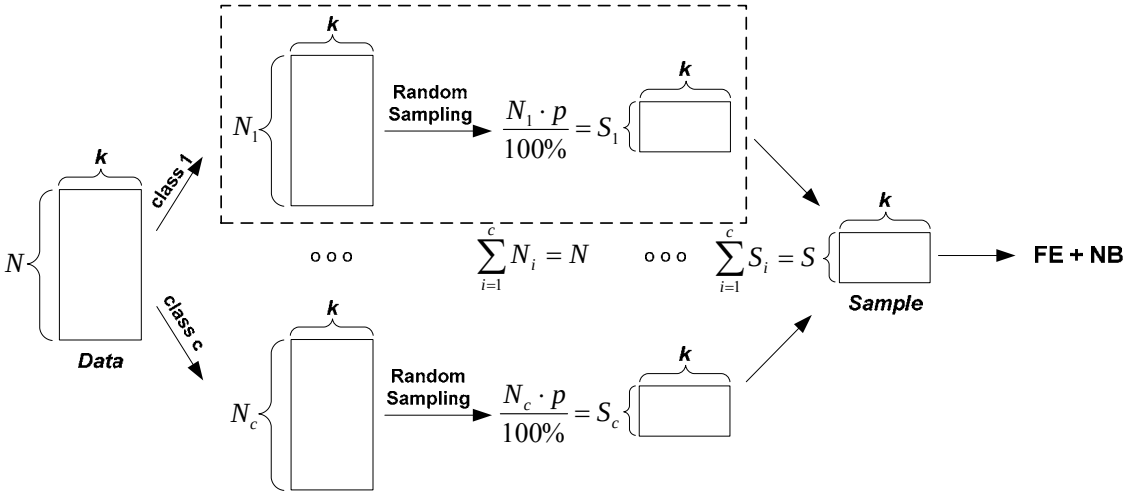

Figure 2. Stratified random sampling.

fined number of instances (called bucket size) or cannot be split further. The order in which features are chosen to split can result in different *kd*-trees. As the goal of partitioning for selective sampling is to split instances into different (dissimilar) groups, a splitting feature is chosen if the data variance is maximized along the dimension associated with the splitting feature.

In Figure 3 (area inside the dashed box) the basic idea of selective sampling is presented graphically. First, a *kd*-tree is constructed from data, then a defined percent of instances is selected from each leaf of the tree and added to the resulting sample to be used by FE and NB models construction.

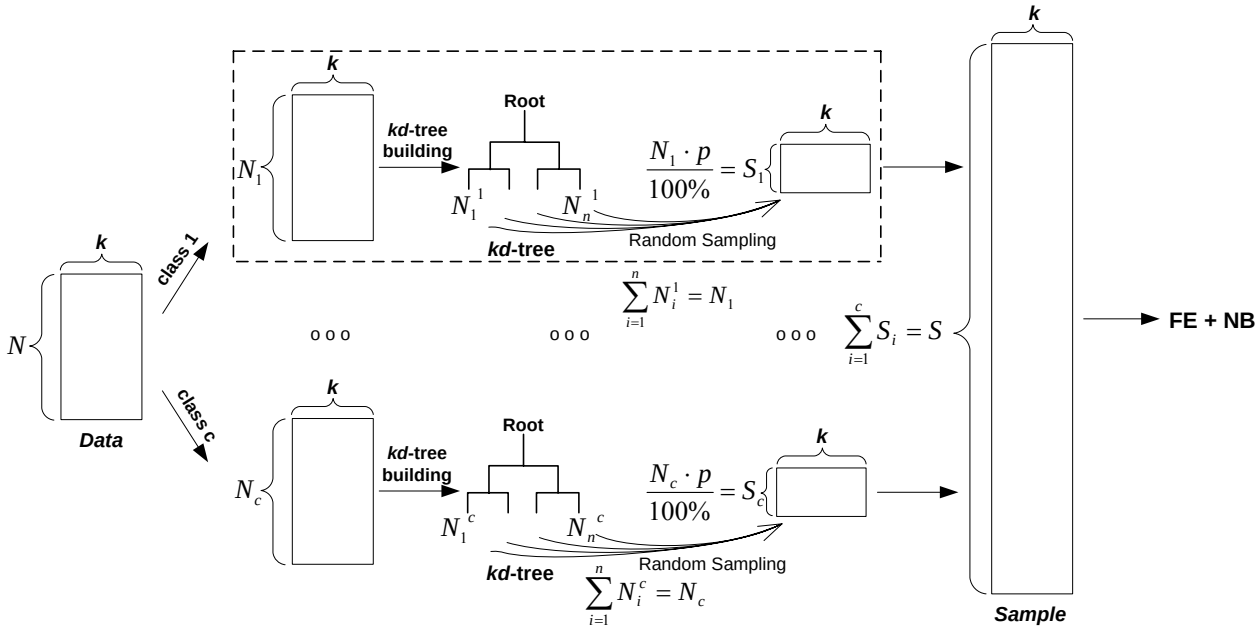

Figure 3. Stratified sampling with *kd*-tree based selection of (representative) instances.

Liu [12] proposed to use *kd*-tree based selective sampling approach for unsupervised feature selection.

We shall run a few steps forward to the analysis of the results of experimental study and say that, although being different in nature, stratified sampling and *kd*-tree based sampling have similar effect with respect to the application of FE for NB classification. This fact is the main motivation to try the combination of these approaches, so that both class information and information about data distribution are used (as presented in Figure 3). It can be seen from the figure that the main difference is in constructing several local *kd*-trees for each group of instances related to certain class, instead of constructing one global tree.

5. EXPERIMENTS AND RESULTS

The experiments were conducted on 10 data sets with different characteristics taken from the UCI machine learning repository [4]. The main characteristics of the data sets are presented in Table 1, which include 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 numerical and categorical/binary features, and total number of numerical plus binary or binarized categorical features 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 NB learning algorithm was calculated. Although NB relies on an assumption that the features used for deriving a prediction are independent of each other, given the predicted value (that is rarely true in practice), it has been recently shown that the NB can be optimal even when the independence assumption is violated by a wide margin [5]. It was shown that the NB can be effectively used in ensemble techniques, which perform also bias reduction, as boosting [6].

Table 1. Datasets characteristics

Dataset	inst	class	Feat (num)	Feat (cat/bin)	Feat (num+bin)
Hypothyr.	3772	3	7	22	31
Ionosph.	351	2	33	0	33
Kr-vs-kp	3196	2	0	37	40
Liver	345	2	6	0	6
Monk-1	432	2	0	6	15
Monk-2	432	2	0	6	15
Monk-3	432	2	0	6	15
Tic	958	2	0	9	27
Vehicle	846	4	18	0	18
Waveform	5000	3	21	0	21

For example, Elkan's application of the boosted NB won the first place out of 45 entries in the data mining competition KDD'97 [6]. Beside this, when the NB is applied to the subproblems of lower dimensionalities as in the random subspace method, the error bias of the Bayesian probability estimates caused by the independence assumption becomes smaller. We can take into consideration also the fact that such FE techniques like PCA are aimed not only to reduce the dimensionality but also to produce

uncorrelated features.

For each data set 30 test runs of Monte-Carlo cross validation were made to evaluate classification accuracy with the four FE approaches and without FE. 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 20 percent instances of the data set are first randomly picked up to the test set. The sampling approaches are applied to the remaining 80 percent instances to form the training set, which is used for finding the feature-extraction transformation matrix w. We were selecting $p = i*10\%$ from the original sample, with $i = \{1, \ldots, 10\}$. The bucket size for a *kd*-tree was selected proportionally for each data set, equal to 10% of original number of instances.

For PCA and parametric FE we used a 0.85 variance threshold, and we took all the features extracted by parametric FE as it was always equal to *number of classes-1*. The test environment was implemented within the WEKA framework (the machine learning library in Java) [17].

In Figure 4 accuracies of NB classification are presented for different sample sizes (from 10% to 100%). The figure is divided into four parts, each presenting the results of one sampling strategy: a) random sampling, b) stratified random sampling, c) *kd*-tree based sampling, and d) stratified *kd*-tree based sampling. For each sampling strategy four approaches are compared: *Plain*

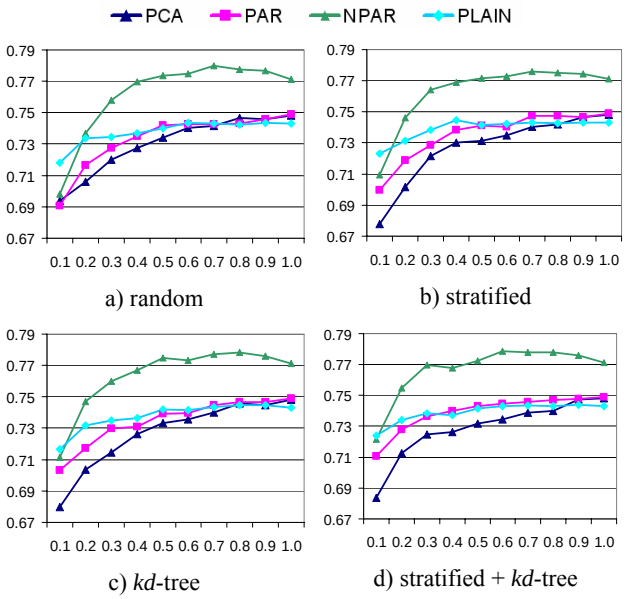

Figure 4. Naïve Bayes accuracy with different samples selected by a) random sampling, b) stratified random sampling, c) *kd*-tree based sampling, and d) stratified *kd*-tree based sampling.

denotes the situation where NB is applied without any preceding FE, and *PCA*, *PAR*, and *NPAR* denote correspondingly the situations when PCA, parametric FE and nonparametric FE are applied for NB.

With random sampling (Figure 4a), NB produces the highest accuracy when no FE is applied (Plain) with $p < 20\%$, and when nonparamentic FE (NPAR) is applied if $p > 20\%$. For $p > 30\%$ *NPAR* results in (at least 2%) higher NB accuracy than *PCA*, *PAR*, and *Plain* even with $p = 100\%$. *NPAR* produces highest accuracy values when $p \approx 70\%$ and in this respect it behaves differently than others which produce highest accuracy values with $p = 100\%$.

PAR achieves the same level as *Plain* when $p = 50\%$ and slightly outperforms *Plain* when $p > 80\%$. *PCA* is the worst when $10\% < p < 70\%$. With $p > 70\%$ *PCA* behaves very similar to *PAR*.

When stratified sampling is applied (Figure 4b), *Plain* is best only when $p \approx 10\%$ and *NPAR* is the best when $p \geq 20\%$. As with random sampling for $p > 30\%$ *NPAR* results in (at least 2%) higher NB accuracy than *PCA*, *PAR*, and *Plain* even with $p = 100\%$. *NPAR* produces highest accuracy values when $p \approx 70\%$. PAR achieves the same level as *Plain* when $p = 50\%$ and slightly outperforms *Plain* when $p > 70\%$. *PCA* is the worst when $p < 80\%$ and achieves the same level of *Plain* when $p \approx 80\%$, and *PAR* when $p \geq 90\%$.

The *kd*-tree based sampling (Figure 4c) has very similar effect to the stratified sampling on the performance of NB, although being different in nature. The only difference is that *NPAR* produces highest accuracy values when $p \approx 80\%$.

The *kd*-tree based sampling when applied for each class of instances separately (Figure 4d), improves the positive effect of stratified sampling wrt each FE for $p < 50\%$. Also, *Plain* and *NPAR* are equal when $p = 10\%$, *NPAR* is the best when $p > 10\%$, and NPAR produces the highest accuracy values when $p \approx 60\%$.

Comparing the results related to four sampling strategies we can conclude that no matter which one of four sampling strategies is used, if sample size is small, $p \approx 10\%$, then *Plain* shows the best accuracy results; if sample size $p \geq 20\%$, then *NPAR* outperforms other methods; and if sample size $p \geq 30\%$, *NPAR* outperforms other methods even if they use 100% of sample. The best p for *NPAR* depends on sampling method: for random and stratified $p = 70\%$, for *kd*-tree $p = 80\%$, and for stratified + *kd*-tree $p = 60\%$. PCA is the worst techniques when applied on a small sample size, especially when stratification or *kd*-tree indexing is used.

Generally, all sampling strategies have similar effect on final classification accuracy of NB for $p > 30\%$. The significant difference in performance is within $10\% \leq p \leq 30\%$ as can be seen from the figure.

The intuitive explanation for this is that when taking very large portion of data, it does not matter which strategy is used since most of the selected instances likely to be the same (maybe chosen in different orders). However, the smaller the portion of the sample, the more important is how the instances are selected.

Figure 5 shows how stratification improves the effect of *kd*-tree sampling for $10\% \leq p \leq 30\%$. The left part of the figure shows the difference in NB accuracy due to use of random sampling comparing to *kd*-tree based sampling, and the right part – due to use of random sampling comparing to *kd*-tree based sampling with stratification.

6. CONCLUSIONS AND FURTHER RESEARCH

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". When a date set includes many instances, sample reduction techniques are

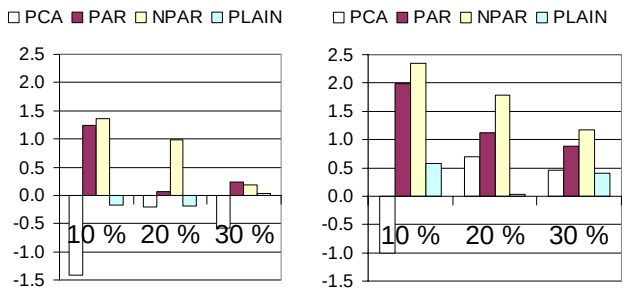

Figure 5. The *kd*-tree based sampling vs. the random sampling (left) and the stratified *kd*-tree based sampling vs. the random sampling (right).

used before applying DM techniques.

In this paper, we analyzed the impact of sample reduction on the process of FE for SL. The experimental results of our study show that the type of sampling approach is not important when the selected sample size is relatively large. However, it is important to take into account both class information and information about data distribution when the sample size to be selected is small. With regard to this, we are planning to analyze further the performance of FE for SL when 5-30% of instances are selected from a data set. Also it might be interesting to see for which data sets there was a significant difference in samples selected by different strategies, e.g. how many instances from random sample are significantly different from instances selected by stratified random sampling or *kd*-tree based selective sampling.

Actual time taken to build classification models with and without FE with regard to the selected samples 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.

It is interesting to analyze the sample reduction impact on the other commonly applied learning algorithms like decision trees and lazy learning and compare results with the reported findings.

7. ACKNOWLEDGMENTS

This research is partly supported by the COMAS Graduate School of the University of Jyväskylä, Finland, and Science Foundation Ireland under Grant No. S.F.I.-02IN.1I111.

8. REFERENCES

[1] Aivazyan, S.A. *Applied statistics: classification and dimension reduction*. Finance and Statistics, Moscow, 1989.

[2] Aladjem, M. Parametric and nonparametric linear mappings of multidimensional data. *Pattern Recognition 24*(6), 1991, 543-553.

[3] Bellman, R., *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.

[4] Blake, C.L., Merz, C.J. UCI *Repository of Machine Learning Databases*. Dept. of Information and Computer Science, University of California, Irvine CA, 1998.

[5] Domingos P. and Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning 29* (2,3), 1997, 103–130.

[6] Elkan C. *Boosting and naïve Bayesian learning*, Tech. Report CS97-557, Department of CS and Engineering, University of California, San Diego, USA, 1997

[7] Fayyad, U.M. Data Mining and Knowledge Discovery: Making Sense Out of Data, *IEEE Expert 11*(5), 1996, 20-25.

[8] Fukunaga, K. *Introduction to statistical pattern recognition*. Academic Press, London, 1990.

[9] Gaede V., Günther O. Multidimensional access methods, *ACM Comput. Surv. 30* (2), 1998, 170–231.

[10] 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.

[11] Jolliffe, I.T. *Principal Component Analysis*. Springer, New York, NY, 1986.

[12] Liu H., Motoda H., Yu L. A selective sampling approach to active feature selection, *Artificial Intelligence 159*(1-2), 2004, 49-74.

[13] Michalski, R.S.. Seeking Knowledge in the Deluge of Facts, *Fundamenta Informaticae* 30, 1997, 283-297.

[14] 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.

[15] Pechenizkiy M. Impact of the Feature Extraction on the Performance of a Classifier: kNN, Naïve Bayes and C4.5. In: B.Kegl, G.Lapalme (Eds.): *Proc. of 18th CSCSI Conference on Artificial Intelligence AI'05*, LNAI 3501, Springer-Verlag, 2005, 268-279.

[16] 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*, LNCS 2798, Springer-Verlag, 2003, 278-292.

[17] Witten I. and Frank E. *Data Mining: Practical machine learning tools with Java implementations*, Morgan Kaufmann, San Francisco, 2000.