Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 2 (June), pp. 317-327

Classification Performance Using Principal Component Analysis and Different Value of the Ratio R

J. Novakovic, S. Rankov

Jasmina Novakovic
"Faculty of Computer Science" Megatrend University Belgrade
Serbia, 11000 Belgrade, Bulevar Umetnosti 29
E-mail: jnovakovic@megatrend.edu.rs

Sinisa Rankov
Megatrend University Belgrade
Bulevar Umetnosti 29
E-mail: rankovs@megatrend.edu.rs

Abstract: A comparison between several classification algorithms with feature extraction on real dataset is presented. Principal Component Analysis (PCA) has been used for feature extraction with different values of the ratio R, evaluated and compared using four different types of classifiers on two real benchmark data sets. Accuracy of the classifiers is influenced by the choice of different values of the ratio R. There is no best value of the ratio R, for different datasets and different classifiers accuracy curves as a function of the number of features used may significantly differ. In our cases feature extraction is especially effective for classification algorithms that do not have any inherent feature selections or feature extraction build in, such as the nearest neighbour methods or some types of neural networks.
Keywords: feature extraction, linear feature extraction methods, principal component analysis, classification algorithms, classification accuracy.

1 Introduction

Data dimensionality reduction is an active field in computer science. It is a fundamental problem in many different areas, especially in forecasting, document classification, bioinformatics, and object recognition or in modelling of complex technological processes. In such applications datasets with thousands of features are not uncommon. All features may be important for some problems, but for some target concept only a small subset of features is usually relevant.

To overcome the curse of dimensionality problem, dimensionality of the feature space should be reduced. This may be done by selecting only the subset of relevant features, or creating new features that contain maximum information about the class label from the original ones. The former methodology is named feature selection, while the latter is called feature extraction, and it includes linear (PCA, Independent Component Analysis (ICA) etc.) and non-linear feature extraction methods. Finding new features subset are usually intractable and many problem related to feature extraction have been shown to be NP-hard ([1]).

Feature extraction brings the immediate effects for application: speeding up a data mining algorithm, improving the data quality and thereof the performance of data mining, and increasing the comprehensibility of the mining results. It has been a fertile field of research and development since 1970's in statistical pattern recognition ([2] and [3]), machine learning and data mining.

Some classification algorithms have inherited ability to focus on relevant features and ignore irrelevant ones. Decision trees are primary example of a class of such algorithms ([4] and [5]), but also multi-layer perceptron (MLP) neural networks with strong regularization of the input

layer may exclude the irrelevant features in an automatic way ([6]). Such methods may also benefit from independent feature selection or extraction. On the other hand, some algorithms have no provisions for feature selection or extraction. The k-nearest neighbour algorithm (k-NN) is one family of such methods that classify novel examples by retrieving the nearest training example, strongly relaying on feature selection or extraction methods to remove noisy features.

2 PCA

PCA is a standard statistical technique that can be used to reduce the dimensionality of a data set. It ([7]) is known as Karhunen-Loeve transform, has proven to be an exceedingly useful tool for dimensionality reduction of multivariate data with many application areas in image analysis, pattern recognition and appearance-based visual recognition, data compression, time series prediction, and analysis of biological data - to mention a few.

The strength of PCA for data analysis comes from its efficient computational mechanism, the fact that it is well understood, and from its general applicability. For example, a sample of applications in computer vision includes the representation and recognition of faces ([8], [9], [10] and [11]), recognition of 3D objects under varying pose ([12]), tracking of deformable objects ([13]), and for representations of 3D range data of heads([14]).

PCA is a method of transforming the initial data set represented by vector samples into a new set of vector samples with derived dimensions. The basic idea can be described as follows: a set of n-dimensional vector samples $X = \{x_1, x_2, x_3, ..., x_m\}$ should be transformed into another set $Y = \{y_1, y_2, ..., y_m\}$ of the same dimensionality, but y-s have the properties that most of their information content is stored in the first few dimensions. So, we can reduce the data set to a smaller number of dimensions with low information loss.

The transformation is based on the assumption that high information corresponds to high variance. If we want to reduce a set of input dimensions X to a single dimension Y, we should transform X into Y as a matrix computation

$$Y = A \cdot X \tag{1}$$

choosing A such that Y has the largest variance possible for a given data set. The single dimension Y obtained in this transformation is called the first principal component. This component is an axis in the direction of maximum variance. The first principal component minimizes the distance of the sum of squares between data points and their projections on the component axis. In practice, it is not possible to determine matrix A directly, and therefore we compute the covariance matrix S as a first step in features transformation. Matrix S ([15]) is defined as

$$S_{n\times n} = \frac{1}{n-1}\sum_{j=1}^{n}(x_j - x')^T \cdot (x_j - x') \tag{2}$$

where

$$x' = \frac{1}{n}\sum_{j=1}^{n} x_j \tag{3}$$

In the next step, the eigenvalues of the covariance matrix S for the given data should be calculated. Finally, the m eigenvectors corresponding to the m largest eigenvalues of S define a linear transformation from the n-dimensional space to an m-dimensional space in which the features are uncorrelated. To specify the principal components we need the following additional explanations about the notation in matrix S: 1) The eigenvalues of $S_{n\times n}$ $\lambda_1, \lambda_2, ..., \lambda_n$, where $\lambda_1 \geq$

$\lambda_2 \geq, ..., \geq \lambda_n \geq 0$ and 2) The eigenvectors $e_1, e_2, ..., e_n$ correspond to eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$, and they are called the principal axes.

Principal axes are new, transformed axes of n-dimensional space, where the new variables are uncorrelated, and variance for the i-th component is equal to the i-th eigenvalue. Most of the information about the data set is concentrated in a few first principal components. In this paper we research how many of the principal components are needed to get a good representation of the data. In other words, we try to find the effective dimensionality of the data set. For this purpose we analyze the proportion of variance. Dividing the sum of the first m eigenvalues by the sum of all the variances (all eigenvalues), we will get the measure for the quality of representation based on the first m principal components. The result is expressed as a percentage. The criterion for features selection is based on the ratio of the sum of the m largest eigenvalues of S to the trace of S. That is a fraction of the variance retained in the m-dimensional space. If the eigenvalues are labeled so that $\lambda_1 \geq \lambda_2 \geq, ..., \geq \lambda_n$, then the ratio can be written as

$$R = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \tag{4}$$

All analyses of the subset of m features represent a good initial estimate of the n-dimensionality space if the ratio R is sufficiently large, it means greater than the threshold value. This method is computationally inexpensive, but it requires characterizing data with the covariance matrix S.

In implementation, the transformation from the original attributes to principal components is carried out through a process by first computing the covariance matrix of the original attributes and then, by extracting its eigenvectors to act as the principal components. The eigenvectors specify a linear mapping from the original attribute space of dimensionality N to a new space of size M in which attributes are uncorrelated.

The resulting eigenvectors can be ranked according to the amount of variation in the original data that they account for. Typically, the first few transformed attributes account for most of the variation in the data set and are retained, while the remainders are discarded.

PCA is an unsupervised method, which makes no use of information embodied within the class variable. Because, the PCA returns linear combinations of the original features, the meaning of the original features is not preserved. Over the years there have been many extensions to conventional PCA. For example, Independent Component Analysis (ICA) ([16] and [17]) is the attempt to extend PCA to go beyond decorrelation and to perform a dimension reduction onto a feature space with statistically independent variables. Other extensions address the situation where the sample data live in a low-dimensional (non-linear) manifold in an effort to retain a greater proportion of the variance using fewer components ([18], [19], [20], [21], [22] and [23]) and yet other (related) extensions derive PCA from the perspective of density estimation (which facilitate modeling non-linearities in the sample data) and the use of Bayesian formulation for modeling the complexity of the sample data manifold ([24]).

3 Classification Algorithms

Four supervised learning algorithms are adopted here to build models, namely, IB1, Naive Bayes, C4.5 decision tree and the radial basis function (RBF) network. This section gives a brief overview of these algorithms.

3.1 IB1

IB1 is nearest neighbour classifier. It uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances have the same (smallest) distance to the test instance, the first one found is used. Nearest neighbour is one of the simplest learning/classification algorithms, and has been successfully applied to a broad range of problems ([25]).

To classify an unclassified vector X, this algorithm ranks the neighbours of X amongst a given set of N data $(X_i, c_i), i = 1, 2, ..., N$, and uses the class labels c_j $(j = 1, 2, ..., K)$ of the K most similar neighbours to predict the class of the new vector X. In particular, the classes of these neighbours are weighted using the similarity between X and each of its neighbours, where similarity is measured by the Euclidean distance metric. Then, X is assigned the class label with the greatest number of votes among the K nearest class labels.

The nearest neighbour classifier works based on the intuition that the classification of an instance is likely to be most similar to the classification of other instances that are nearby to it within the vector space. Compared to other classification methods such as Naive Bayes', nearest neighbour classifier does not rely on prior probabilities, and it is computationally efficient if the data set concerned is not very large. If, however, the data sets are large (with a high dimensionality), each distance calculation may become quite expensive. This reinforces the need for employing PCA and information gain-based feature ranking to reduce data dimensionality, in order to reduce the computation cost.

3.2 Naive Bayes

This classifier is based on the elementary Bayes' Theorem. It can achieve relatively good performance on classification tasks ([26]). Naive Bayes classifier greatly simplifies learning by assuming that features are independent given the class variable. More formally, this classifier is defined by discriminant functions:

$$f_i(X) = \prod_{j=1}^{N} P(x_j \mid c_i) P(c_i) \tag{5}$$

where $X = (x_1, x_2, ..., x_N)$ denotes a feature vector and $c_j, j = 1, 2, ..., N$, denote possible class labels.

The training phase for learning a classifier consists in estimating conditional probabilities $P(x_j \mid c_i)$ and prior probabilities $P(c_i)$. Here, $P(c_i)$ are estimated by counting the training examples that fall into class c_i and then dividing the resulting count by the size of the training set. Similarly, conditional probabilities are estimated by simply observing the frequency distribution of feature x_j within the training subset that is labeled as class c_i. To classify a class-unknown test vector, the posterior probability of each class is calculated, given the feature values present in the test vector; and the test vector is assigned to the class that is of the highest probability.

3.3 C4.5 Decision Tree

Different methods exist to build decision trees, but all of them summarize given training data in a tree structure, with each branch representing an association between feature values and a class label. One of the most famous and representative amongst these is the C4.5 tree ([27]). The C4.5 tree works by recursively partitioning the training data set according to tests on the

potential of feature values in separating the classes. The decision tree is learned from a set of training examples through an iterative process, of choosing a feature and splitting the given example set according to the values of that feature. The most important question is which of the features is the most influential in determining the classification and hence should be chosen first. Entropy measures or equivalently, information gains are used to select the most influential, which is intuitively deemed to be the feature of the lowest entropy (or of the highest information gain). This learning algorithm works by: a) computing the entropy measure for each feature, b) partitioning the set of examples according to the possible values of the feature that has the lowest entropy, and c) for each are used to estimate probabilities, in a way exactly the same as with the Naive Bayes approach. Note that although feature tests are chosen one at a time in a greedy manner, they are dependent on results of previous tests.

3.4 RBF Networks

A popular type of feed forward network is RBF network. Usually, the RBF network consists of three layers, i.e., the input layer, the hidden layer with Gaussian activation functions, and the output layer. Each hidden unit essentially represents a particular point in input space, and its output, or activation, for a given instance depends on the distance between its point and the instance-which is just another point. Intuitively, the closer these two points, the stronger the activation. This is achieved by using a nonlinear transformation function to convert the distance into a similarity measure. A bell-shaped Gaussian activation function, whose width may be different for each hidden unit, is commonly used for this purpose. The hidden units are called RBFs because the points in instance space for which a given hidden unit produces the same activation form a hypersphere or hyperellipsoid.

The output layer of an RBF network takes a linear combination of the outputs of the hidden units and-in classification problems-pipes it through the sigmoid function. The parameters that such a network learns are (a) the centers and widths of the RBFs and (b) the weights used to form the linear combination of the outputs obtained from the hidden layer.

One way to determine the first set of parameters is to use clustering, without looking at the class labels of the training instances at all. The simple k-means clustering algorithm can be applied, clustering each class independently to obtain k basis functions for each class. Intuitively, the resulting RBFs represent prototype instances. Then the second set of parameters can be learned, keeping the first parameters fixed. This involves learning a linear model using one of the techniques such as, linear or logistic regression. If there are far fewer hidden units than training instances, this can be done very quickly.

RBF networks ([28]) are a special class of neural networks in which the distance between the input vector and a prototype vector determines the activation of a hidden neuron. Prototype vectors refer to centers of clusters obtained during RBF training. Usually, three kinds of distance metrics can be used in this network, such as Euclidean, Manhattan, and Mahalanobis distances. The RBF network provides a function $Y : R^n \rightarrow R^M$, which maps n-dimensional input patterns to M-dimensional outputs ($\{(X_i, Y_i) \in R^n \times R^M, i = 1, 2, ..., N\}$). Assume that there are M classes in the data set. The m-th output of the network is as follows ([29]):

$$y_m(X) = \sum_{j=1}^{K} w_{m_j} \theta_j(X) + w_{m_0} b_m \tag{6}$$

In this case X is the n-dimensional input pattern vector, $m = 1, 2, ..., M$, and K is the number of hidden units. M is the number of classes (outputs), w_{m_j} is the weight connecting the j-th

hidden unit to the m-th output node, b_m is the bias, and w_{m_0} is the weight connecting the bias and the m-th output node.

The radial basis activation function $\theta(x)$ of the RBF network distinguishes it from other types of neural networks. Several forms of activation functions have been used in applications ([29]):

- $\theta(x) = e^{\frac{-x^2}{2\sigma^2}}$ (7)
- $\theta(x) = (x^2 + \sigma^2)^{-\beta}, \beta > 0$ (8)
- $\theta(x) = (x^2 + \sigma^2)^{\beta}, \beta > 0$ (9)
- $\theta(x) = x^2 \ln(x)$ (10)

here σ is a parameter that determines the smoothness properties of the interpolating function. A disadvantage of RBF networks is that they give every feature the same weight because all are treated equally in the distance computation. Hence they cannot deal effectively with irrelevant features.

4 Experiments and results

Real datasets called "Statlog (Australian credit approval)" and "Statlog (German credit data)" for tests were used, taken from the UCI repository of machine learning databases. These datasets were used to compare the classification performance using IB1, Naive Bayes, RBF networks and C4.5 decision tree classifiers, in conjunction with the use of PCA and different value of the ratio R. The classification performance is measured using ten-foldcross-validation.

German Credit Data

This dataset classifies people described by a set of features as good or bad credit risks. Data set characteristics is multivariate, feature characteristics are categorical and integer. Number of instances is 1000, number of features is 20, and there are no missing values.

Australian Credit Approval

This file concerns credit card applications. Data set characteristics is multivariate; feature characteristics are categorical, integer and real. Number of instances is 690, number of features is 14, and there are missing values. This dataset is interesting because there is a good mix of features continuous, nominal with small numbers of values, and nominal with larger numbers of

values. There are also a few missing values.

Australian credit approval data	Classification accuracy without PCA
Naive Bayes	77,7
C4.5 decision tree	86,1
IB1 classifier	81,2
RBF network	79,7

Table 1: Classification results for Australian credit approval data without using PCA.

The number of input components produced - Australian credit approval data	20	22	25	29	29	30	32	33
R	0,8	0,85	0,9	0,95	0,96	0,97	0,98	0,99

Table 2: The number of input components produced using PCA at various ratio R values for Australian credit approval data.

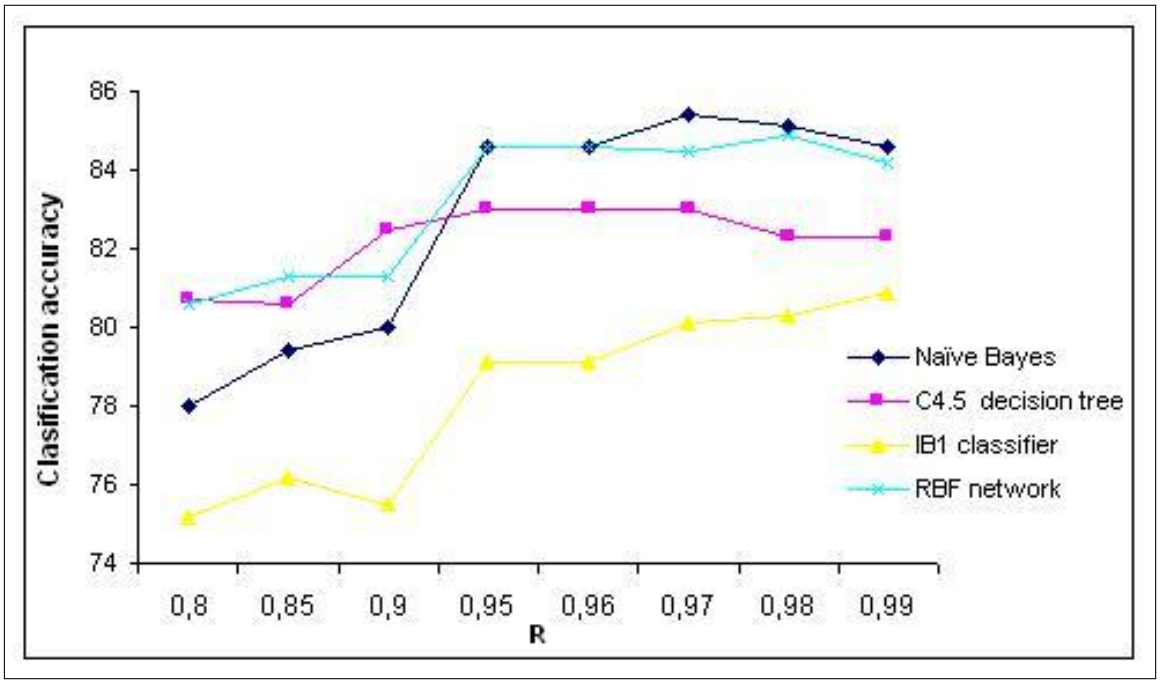

Figure 1: The classification performance using Naive Bayes, C4.5 decision tree, IB1 and RBF network classifiers, in conjunction with the use of PCA and different value of the ratio R. Statlog (Australian credit approval) data set

Classification results without using PCA as a standard statistical technique that can be used to reduce the dimensionality of a data set, for Australian credit approval are presented in Table 1, and for German credit data in Table 3.

Table 2 and 4 show the number of input components produced for each ratio R value investigated. It can be observe from the tables that the number of input components reduces with decreasing values of the ratio R used.

In Figure 1 classification performance for Naive Bayes and RBF network are significantly better with PCA. For IB1 and C4.5 decision tree classifiers the results are better without PCA.

For greater value of the ratio R classification accuracy of IB1 classifier is better. The others types of classifier in our experiment have the better results with greater value, but when value reached some boundaries performance of classifier are the same or worst.

German credit data	Classification accuracy without PCA
Naive Bayes	75,4
C4.5 decision tree	70,5
IB1 classifier	72
RBF network	74

Table 3: Classification results for German credit data without using PCA.

The performance of some classifiers depends on its generalization capability, which in turn is dependent upon the data representation. One important characteristic of data representation is uncorrelated. This is because correlated data reduce the distinctiveness of data representation and thus, introduce confusion to the classifier model during the learning process and hence, producing one that has low generalization capability to resolve unseen data. The results demonstrated that the elimination of correlated information in the sample data by way of the PCA method improved Naive Bayes and RBF network classification performance (Figure 1).

At point 0.95% of the ratio R value with 29 input components, all classifiers significantly improved the classification accuracy. After that the ratio R value, the classification accuracy is about the same with little variations between classifiers. It suggests that this number of inputs is sufficiently optimal for the all classifiers to learn distinct features in the data and perform better input/output mapping.

The number of input components produced - German credit data	31	34	38	42	43	44	45	46
R	0,8	0,85	0,9	0,95	0,96	0,97	0,98	0,99

Table 4: The number of input components produced using PCA at various ratio R values for German credit data.

German credit data doesn't consist of correlated information caused by overlapping input instances. Without correlation in sampled data there is not confusion in classifiers during the learning process (Figure 2) and thus, no degrades their generalization capability. In this case all classifiers' classification performance doesn't improved by PCA. For greater value of the ratio R classification accuracy of IB1 classifier is better. Naive Bayes classifier has the worst results with greater values of the ratio R. Classification accuracy of RBF network have oscillation values. Classification accuracy for C4.5 decision tree doesn't change too much with different values of the ratio R.

This final part of the comparative study is set to investigate the differences between different classifiers, in terms of their classification ability. It is clear from Figures 1 and 2 that on average, Naive Bayes and RBF network classifiers tend to significantly outperform the decision tree and IB1 classifiers.

5 Conclusions

Feature extraction leading to reduced dimensionality of the feature space. PCA is one of the most popular techniques for dimensionality reduction of multivariate data points with application

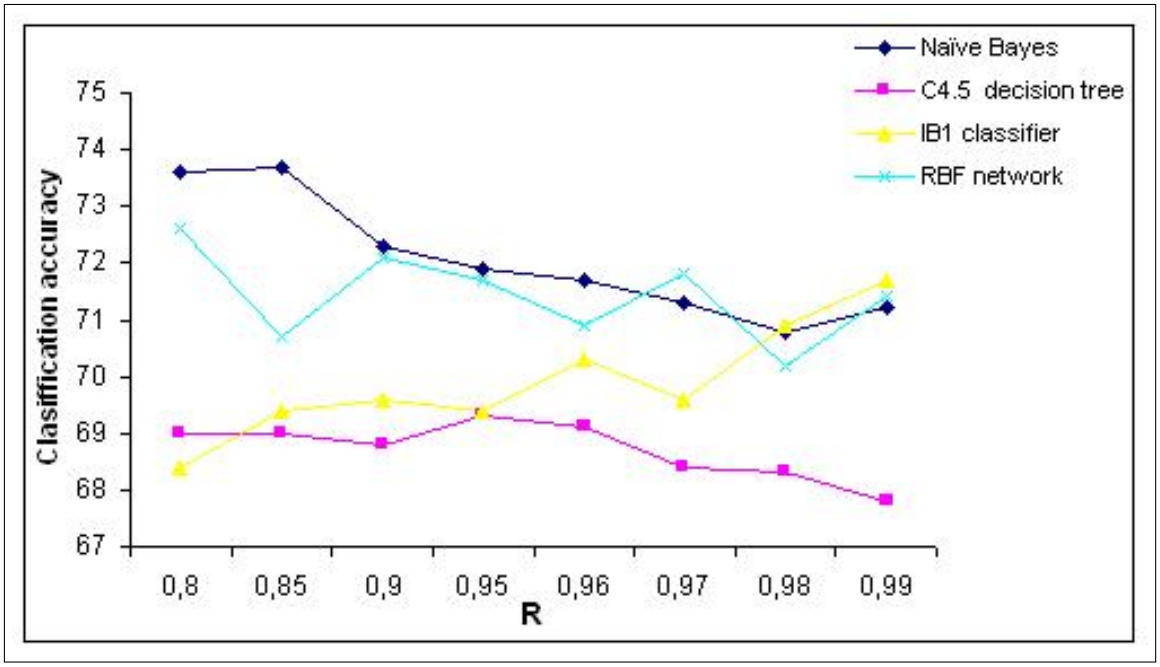

Figure 2: The classification performance using Naive Bayes, C4.5 decision tree, IB1 and RBF network classifiers, in conjunction with the use of PCA and different value of the ratio R. Statlog (German credit data) data set

areas covering many branches of science. This is especially effective for classification algorithms that do not have any inherent feature selections or feature extraction builds in, such as the nearest neighbour methods or some types of neural networks. PCA has been used for feature extraction with different values of the ratio R, evaluated and compared using four different types of classifiers on two real benchmark data sets. Accuracy of the classifiers is influenced by the choice different values of ratio R (Figure 1 and Figure 2).

There is no best value of the ratio R, for different datasets and different classifiers accuracy curves as a function of the number of features used may significantly differ. But, in more cases, the value of 0.95 gave the best results.

Several improvements of the feature extraction method presented here are possible:

- The algorithms and datasets will be selected according to precise criteria: different algorithms with PCA as linear feature extraction method, and several datasets, either real or artificial, with nominal, binary and continuous features.
- ICA and others linear feature extraction methods may be included.
- Problem of data dimensionality reduction may be analysed with non-linear feature extraction methods.

These conclusions and recommendations will be tested on larger datasets using various classification algorithms in the near future.

Bibliography

[1] A.L. Blum, R.L. Rivest, Training a 3-node neural networks is NP-complete, *Neural Networks,* 5:117 - 127, 1992.

[2] N. Wyse, R. Dubes, A.K. Jain, A critical evaluation of intrinsic dimensionality algorithms. In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice,* pp 415–425. Morgan Kaufmann Publishers, Inc., 1980.

[3] M. Ben-Bassat, Pattern recognition and reduction of dimensionality, In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of statistics-II,* pp 773-791. North Holland, 1982.

[4] L.Breiman, J.H. Friedman, R.H. Olshen, Stone C.J., *Classification and Regression Trees,* Wadsworth and Brooks, Monterey, CA, 1984.

[5] J.R. Quinlan, *C4.5: Programs for machine learning,* San Mateo, Morgan Kaufman, 1993.

[6] W. Duch, R. Adamczak, K. Grabczewski, *A new methodology of extraction, optimization and application of crisp and fuzzy logical rules,* IEEE Transactions on Neural Networks, vol. 12, pp. 277-306, 2001.

[7] I.T. Jolliffe, *Principal Component Analysis, Springer-Verlag,* New York, 1986.

[8] L. Sirovich, M. Kirby, Low dimensional procedure for the characterization of human faces, *Journal of the Optical Society of America,* 4(3) 519-524, 1987.

[9] M. Turk, A. Pentland, Eigen faces for recognition, *J. of Cognitive Neuroscience 3(1),* 1991.

[10] B. Moghaddam, A. Pentland, B. Starner, View-based and modular eigenspaoes for face recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp 84-91, 1994.

[11] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection, *Proceedings of the European Conference on Computer Vision,* 1996.

[12] H. Murase, S.K. Nayar, Learning and recognition of 3D objects from appearance, *IEEE 2nd Qualitative Vision Workshop,* pp 39-50, New York, NY, June 1993.

[13] M. J. Black, D. Jepson, Eigen-tracking: Robust matching and tracking of articulated objects using a view-based representation, *Proceedings of the European Conference on Computer Vision (ECCV),* pp 329-342, Cambridge, England, 1996.

[14] J.J. Atick, P.A. Griffin, N.A. Redlich, Statistical approach to shape-from-shading: deriving 3d face surfaces from single 2d images, *Neural Computation,* 1997.

[15] M. Kantardzic,*Data Mining: Concepts, Models, Methods, and Algorithms,* John Wiley & Sons, 2003.

[16] P. Comon, Independent component analysis, a new concept? *Signal processing pages 36(3),* pp 11-20, 1994.

[17] A.J. Bell, T.J. Sejnowski, An information maximization approach to blind separation and blind deconvolution, *Neural Computation,* pp 1129-1159, 1995.

[18] C. Bregler, S.M. Omohundro, Nonlinear manifold learning for visual speech recognition, *iccv,* Boston, Jun 1995.

[19] T. Heap, D. Hogg, Wormholes in shape space: Tracking through discontinuous changes in shape, *iccv,* 1998.

[20] T. Hastie, W. Stuetzle, Principal curves, *Journal of Americam Statistical Association 84,* pp 502-516, 1989.

[21] M.A. Kramer, Non linear principal component analysis using autoassociative neural networks, *AI Journal 37(2),* pp 233-243, 1991.

[22] A.R. Webb, An approach to nonlinear principal components-analysis using radially symmetrical kernel functions, *Statistics and computing 6(2),* pp 159-168, 1996.

[23] V. Silva, J.B. Tenenbaum, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science,* 290, December 2000.

[24] J.M. Winn, C.M. Bishop, Non-linear bayesian image modelling, *Proceedings of the European Conference on Computer Vision,* Dublin, Ireland, June 2000.

[25] M. Kuramochi, G. Karypis. Gene classification using expression profiles: a feasibility study, *International Journal on Artificial Intelligence Tools,* 14(4):641-660, 2005.

[26] P. Domingos, M. Pazzani, Feature selection and transduction for prediction of molecular bioactivity for drug design, *Machine Learning,* 29:103-130, 1997.

[27] E. P. Xing, M. L. Jordan, R. M. Karp Feature selection for high-dimensional genomic microarray data, *Proceedings of the 18th International Conference on Machine Learning,* 601-608, 2001.

[28] C.M. Bishop, *Neural Network for Pattern Recognition,* Oxford University Press Inc., New York, 1995.

[29] L. Wang, X. Fu, *Data Mining with Computational Intelligence,* Springer-Verlag Berlin Heidelberg, Germany, pages 9-14, 2005.