# Nonlinear Feature Transforms Using Maximum Mutual Information

Kari Torkkola
Motorola Labs, 7700 South River Parkway, MD ML28
Tempe AZ 85284, USA, email: kari.torkkola@motorola.com

**Abstract**

*Finding the right features is an essential part of a pattern recognition system. This can be accomplished either by selection or by a transform from a larger number of "raw" features. In this work we learn non-linear dimension reducing discriminative transforms that are implemented as neural networks, either as radial basis function networks or as multilayer perceptrons. As the criterion, we use the joint mutual information (MI) between the class labels of training data and transformed features. Our measure of MI makes use of Renyi entropy as formulated by Principe et al. Resulting low-dimensional features enable a classifier to operate with less computational resources and memory without compromising the accuracy.*

## 1 Introduction

Feature selection or feature transforms are important aspects of any pattern recognition system. Optimal feature selection coupled with a pattern recognition system leads to a combinatorial problem since all combinations of available features need to be evaluated, by actually training and evaluating a classifier. This is called the *wrapper* configuration [10]. Obviously wrapper strategy does not allow to learn feature transforms, because all possible transforms cannot be enumerated.

Another approach is to evaluate some criterion related to the final classification error that would reflect the "importance" of a feature or a number of features jointly. This is called the *filter* configuration in feature selection [10]. What would be an optimal criterion for this purpose? Such a criterion would naturally reflect the Bayes error rate. Approximations to the Bayes error rate can be used, based on Bhattacharyya bound or an interclass divergence criterion. These are usually accompanied by a parametric estimation, such as Gaussian, of the densities at hand [6, 16]. Other criteria and transform implemetations are presented, for example, in [9, 20, 11, 12]

Another such criterion is the joint mutual information (MI) between the features and the class labels [1, 21, 19, 17]. It can be shown that MI minimizes the lower bound of the classification error [4, 14, 18]. However, MI according to Shannon's definition is computationally expensive. Evaluation of the joint MI of a number of variables is plausible through histograms, but only for a few variables [21]. Principe et al showed in [5, 15, 14] that using Renyi's entropy instead of Shannon's, combined with Parzen density estimation, leads to expressions of mutual information with significant computational savings. They explored the MI between two continuous spaces. In an earlier work [18], we extended this method to mutual information between continuous variables and discrete class labels in order to learn linear dimension-reducing linear feature transforms for pattern recognition.

In this paper we apply a Renyi entropy based mutual information measure to learn *non-linear* dimension-reducing transforms. We introduce the mutual information measure based on Renyi's entropy, and describe its application to both Radial Basis Function networks (RBF) and Multilayer Perceptrons (MLP). Finally we describe experiments with several public domain databases.

## 2 Shannon's Definition of Mutual Information

We denote labeled samples of continuous-valued random variable $Y$ as pairs $\{\boldsymbol{y}_i, c_i\}$, where $\boldsymbol{y}_i \in R^d$, and class labels are samples of a discrete-valued random variable $C$, $c_i \in \{1, 2, ..., N_c\}, i \in [1, N]$.

If we draw one sample of $Y$ at random, the entropy or uncertainty of the class label, making use of Shannon's definition, is defined in terms of class prior probabilities

$$H(C) = -\sum_c P(c) \log(P(c)). \quad (1)$$

After having observed the feature vector $\boldsymbol{y}$, our uncertainty of the class identity is the conditional entropy

$$H(C|Y) = \int_{\boldsymbol{y}} p(\boldsymbol{y}) \left( \sum_c p(c|\boldsymbol{y}) \log(p(c|\boldsymbol{y})) \right) d\boldsymbol{y}. \quad (2)$$

The amount by which the class uncertainty is reduced after having observed the feature vector $\boldsymbol{y}$ is called the mutual

information, $I(C,Y) = H(C) - H(C|Y)$, which can be written as

$$I(C,Y) = \sum_c \int_{\boldsymbol{y}} p(c,\boldsymbol{y}) \log \frac{p(c,\boldsymbol{y})}{P(c)p(\boldsymbol{y})} d\boldsymbol{y} \qquad (3)$$

after applying the identities $p(c,\boldsymbol{y}) = p(c|\boldsymbol{y})p(\boldsymbol{y})$ and $P(c) = \int_{\boldsymbol{y}} p(c,\boldsymbol{y})d\boldsymbol{y}$.

Mutual information also measures independence between two variables, in this case between $C$ and $Y$. It equals zero when $p(c,\boldsymbol{y}) = P(c)p(\boldsymbol{y})$, that is, when the joint density of $C$ and $Y$ factors (the condition for independence). Mutual information can thus also be viewed as the divergence between the joint density of the variables, and the product of the marginal densities.

Connection between mutual information and optimal classification is given by Fano's inequality [4]. This result, originating from digital communications, determines a lower bound to the probability of error when estimating a discrete random variable $C$ from another random variable $Y$.

$$Pr(c \neq \hat{c}) \geq \frac{H(C|Y) - 1}{\log(N_c)} = \frac{H(C) - I(C,Y) - 1}{\log(N_c)}, \qquad (4)$$

where $\hat{c}$ is the estimate of $C$ after observing a sample of $Y$, which can be scalar or multivariate. Thus the lower bound on error probability is minimized when the mutual information between $C$ and $Y$ is maximized, or, finding such features achieves the lowest possible bound to the error of a classifier. Whether this bound can be reached or not, depends on the goodness of the classifier.

## 3 A Definition Based on Renyi's Entropy

Instead of Shannon's entropy we apply Renyi's quadratic entropy as described in [14, 18] because of its computational advantages. For a continuous variable $Y$ Renyi's quadratic entropy is defined as

$$H_R(Y) = -\log \int_{\boldsymbol{y}} p(\boldsymbol{y})^2 d\boldsymbol{y} \qquad (5)$$

It turns out that Renyi's measure combined with Parzen density estimation method using Gaussian kernels provides significant computational savings, because a convolution of two Gaussians is a Gaussian.

If the density $p(\boldsymbol{y})$ is estimated as a sum of symmetric Gaussians each centered at a sample $\boldsymbol{y}_i$ as

$$p(\boldsymbol{y}) = \frac{1}{N} \sum_{i=1}^{N} G(\boldsymbol{y} - \boldsymbol{y}_i, \sigma I), \qquad (6)$$

then it follows that the integral in (5) equals

$$\begin{aligned} &\int_{\boldsymbol{y}} p(\boldsymbol{y})^2 d\boldsymbol{y} = \\ &= \frac{1}{N^2} \int_{\boldsymbol{y}} \left( \sum_{k=1}^{N} \sum_{j=1}^{N} G(\boldsymbol{y} - \boldsymbol{y}_k, \sigma I) G(\boldsymbol{y} - \boldsymbol{y}_j, \sigma I) \right) d\boldsymbol{y} \\ &= \frac{1}{N^2} \sum_{k=1}^{N} \sum_{j=1}^{N} G(\boldsymbol{y}_k - \boldsymbol{y}_j, 2\sigma I). \end{aligned} \qquad (7)$$

Thus Renyi's quadratic entropy can be computed as a sum of local interactions as defined by the kernel, over all pairs of samples.

In order to use this convenient property, a measure of mutual information needs to be derived making use of quadratic functions of the densities. Principe et al derive quadratic distance measures for probability density functions somewhat heuristically. First, they consider some known inequalities for L2 distance measure between vectors in $R^D$, and then write analoguous expressions for the divergence between the two densities.

The difference of vectors inequality

$$(\boldsymbol{x} - \boldsymbol{y})^t (\boldsymbol{x} - \boldsymbol{y}) \geq 0 \Leftrightarrow ||\boldsymbol{x}||^2 + ||\boldsymbol{y}||^2 - 2\boldsymbol{x}^t\boldsymbol{y} \geq 0 \qquad (8)$$

leads to the following expression of quadratic divergence

$$K_T(f,g) = \int f(\boldsymbol{x})^2 d\boldsymbol{x} + \int g(\boldsymbol{x})^2 d\boldsymbol{x} - 2\int f(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x} \qquad (9)$$

Since mutual information is expressed as the divergence between the joint density and the product marginals, we can insert them into the quadratic divergence expression, which for two continuous variables leads to

$$\begin{aligned} I_T(Y_1,Y_2) = &\iint p(\boldsymbol{y}_1,\boldsymbol{y}_2)^2 d\boldsymbol{y}_1 d\boldsymbol{y}_2 \\ &+ \iint p(\boldsymbol{y}_1)^2 p(\boldsymbol{y}_2)^2 d\boldsymbol{y}_1 d\boldsymbol{y}_2 \\ &-2 \iint p(\boldsymbol{y}_1,\boldsymbol{y}_2) p(\boldsymbol{y}_1) p(\boldsymbol{y}_2) d\boldsymbol{y}_1 d\boldsymbol{y}_2 \end{aligned} \qquad (10)$$

Between a discrete variable $C$ and a continuous variable $Y$ we have

$$\begin{aligned} I_T(C,Y) = &\sum_c \int_{\boldsymbol{y}} p(c,\boldsymbol{y})^2 d\boldsymbol{y} + \sum_c \int_{\boldsymbol{y}} p(c)^2 p(\boldsymbol{y})^2 d\boldsymbol{y} \\ &- 2\sum_c \int_{\boldsymbol{y}} p(c,\boldsymbol{y}) p(c) p(\boldsymbol{y}) d\boldsymbol{y} \end{aligned} \qquad (11)$$

## 4 Maximizing MI to Learn Feature Transforms

Given a set of training data $\{\boldsymbol{x}_i,\ c_i\}$ as samples of a continuous-valued random variable $X$, $\boldsymbol{x}_i \in R^D$, and class labels as samples of a discrete-valued random variable $C$, $c_i \in \{1, 2, ..., N_c\}, i \in [1, N]$, the objective is to find a transformation to $\boldsymbol{y}_i \in R^d, d \leq D$ such that $\boldsymbol{y}_i = g(\boldsymbol{w}, \boldsymbol{x}_i)$ (or its parameters $\boldsymbol{w}$) that maximizes $I(C, Y)$ the mutual information (MI) between transformed data $Y$ and class labels $C$. The procedure is depicted in Fig. 1.

To this end we need to express $I$ as a function of the data set, $I(\{\boldsymbol{y}_i, c_i\})$, in a differentiable form. Once that is done, we can perform gradient ascent on $I$ as follows

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \eta \frac{\partial I}{\partial \boldsymbol{w}} = \boldsymbol{w}_t + \eta \sum_{i=1}^{N} \frac{\partial I}{\partial \boldsymbol{y}_i} \frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{w}}. \qquad (12)$$

The expressions for the former factor inside the sum in (12), $\partial I / \partial \boldsymbol{y}_i$, and for the actual objective function $I(\{\boldsymbol{y}_i, c_i\})$ can be readily derived using now the Parzen window method with Gaussian kernels as the nonparametric density estimator. This derivation has been presented in [18].

Mutual information $I(\{\boldsymbol{y}_i, c_i\})$ can be interpreted as an *information potential* induced by samples of data of different classes. Partial $\partial I / \partial \boldsymbol{y}_i$ can accordingly be interpreted as an *information force* that other samples exert to sample $\boldsymbol{y}_i$. The three components of the sum in quadratic mutual information (11) give rise to the following three components of the information force: 1) samples within the same class attract each other, 2) all samples regardless of class attract each other, and 3) samples of different classes repel each other. This force, coupled with the latter factor inside the sum in (12), $\partial \boldsymbol{y}_i / \partial \boldsymbol{w}$, tends to change the transform in such a way that the samples in the transformed space move into the direction of the force, and thus increase the MI criterion $I(\{\boldsymbol{y}_i, c_i\})$.

Since only the latter factor in (12), $\partial \boldsymbol{y}_i / \partial \boldsymbol{w}$, is determined by the chosen transformation, the resulting method is a very general procedure that can be applied to any differentiable parametrized transforms. In this paper we aply it to two classes of neural networks, RBFs and MLPs, as the transforms.

Naturally, any better (2nd order) optimization techniques such as conjugate gradients or Levenberg-Marquardt can be applied instead of plain gradient ascent.

## 5 Learning Nonlinear Transforms

Feature transforms are implemented in this work as Radial Basis Function networks or Multilayer Perceptrons. The only difference to learning linear feature transforms as presented in [18] is the latter factor in (12) $\partial \boldsymbol{y}_i / \partial \boldsymbol{w}$. For an MLP this can be computed by backpropagation.

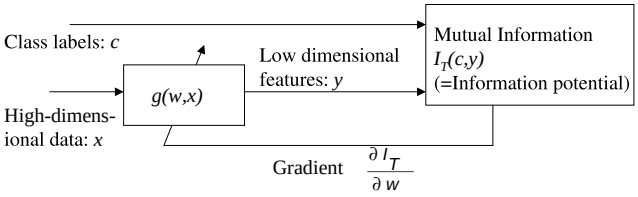


**Figure 1:** Learning feature transforms by maximizing the mutual information between class labels and transformed features.

### 5.1 Multilayer Perceptrons

As an example, let us look at a simple two-layer network with $\boldsymbol{x}$ as the input. The output of the hidden layer $\boldsymbol{z} = g_1(\boldsymbol{W_1 x})$. Transformed data is the output of the second layer $\boldsymbol{y} = g_2(\boldsymbol{W_2 z})$. The hidden layer transfer function $g_1$ is typically hyperbolic tangent, while the output layer transfer function $g_2$ can be either linear or hyperbolic tangent for transform purposes. For example, for a linear output layer ignoring biases, the required gradients become

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{W_2}} = \boldsymbol{z}^T \qquad (13)$$

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{W_1}} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} g_1\prime(\boldsymbol{W_1 x}) \boldsymbol{x}^T \qquad (14)$$

where $\partial \boldsymbol{y} / \partial \boldsymbol{z} = \boldsymbol{W_2}$.

Design parameters for the user to select are thus the number of hidden layer neurons, the type of activation function for the output layer, and the Parzen estimation kernel width, or its annealing schedule. The first should be determined by the complexity of the class distribution structure and the amount of training data. Unfortunately there is no simple principled way of doing this, so several choices probably need to be tried.

The latter two issues are intertwined. If the output layer activation function is hyperbolic tangent, the output is obviously restricted to lie inside a hypercube, and for low dimensions, the kernel width can safely be chosen as a fixed fraction of the cube side length, such as $\sigma = 0.5$. Another simple choice is to choose width so that the resulting kernel function fills a fixed fraction of the volume of the hypercube. However, using a linear activation function with any fixed kernel width merely scales the data of the hidden layer either up or down so that the three components of the information forces are in balance. It is necessary to restrict the output layer weights to rotations only, or equivalently, require the weight vectors to be orthonormal, just as in the case of learning linear transforms [18].

Another issue is the initialization of the weights. A good starting point obviously helps a lot in optimization. We used Linear Discriminant Analysis (LDA) [13] for the first layer, scaling the weights so that some nonlinearity will be used, and LDA again for the second layer. Scaled conjugate gradient optimization was mostly used. Second order methods that explicitly compute an approximation of the Hessian, such as Levenberg-Marquardt, appeared to be too costly with largish networks.

### 5.2 Radial Basis Function Networks

For an RBF the task of computing $\partial \boldsymbol{y}_i / \partial \boldsymbol{w}$, the derivative of the outpur with respect to the parameters of the network, is nothing but standard gradient calculation, and we will not present it here. See, for example [2], page 190, or [7], Table 5.4. Instead, we discuss design parameters and training issues.

The most important design parameters are the number and type (or complexity) of the hidden units. As we had no means of selecting these in a principled fashion, the number of hidden units was simply kept proportional to the number of training examples. Our initialization scheme was to use the Expectation-Maximization algorithm to learn a number of Gaussian basis functions with diagonal covariances separately for each class, rather than for all of the data regardless of class. This appeared to work better as the aim is the class separation. The linear output layer of the RBF was initialized using LDA, and as with MLPs, was restricted to rotations.

A normal design procedure for RBF networks is unsupervised training of the basis functions followed by supervised training of the linear output weights. We experimented by training only the linear part using the MMI criterion keeping the basis functions fixed after the EM algorithm. This resulted in no loss or minimal loss of accuracy compared to full supervised training of all network parameters using MMI. Since the computation involved in the latter is not insignificant, all experiments reported in this paper are run with the former configuration.

In addition, direct connections from the input to the output were added, which improved performance. This is also included in all reported experiments. For an illustration of an RBF network learning to transform three-dimensional features into two see website [1].

## 6 Pattern Recognition Experiments

We repeated the pattern recognition experiments done in [18], now using RBFs and MLPs as the dimension-reducing nonlinear transforms. Five data sets were used, each very different in terms of dimensionality, number of classes, and the amount of data. The sets and some of their characteristics are presented in Table 1. The Phoneme set is available with the LVQ_PAK,[2] and the Pipeline Flow set is available from the Aston University.[3] The rest of the data sets are from the UCI Machine Learning Repository [4] [3].

**Table 1:** Characteristics of the data sets used in classification experiments.

| Data set | Dim. | Classes | Train size | Test size |
|---|---|---|---|---|
| Letter | 16 | 26 | 16000 | 4000 |
| Landsat | 36 | 6 | 4435 | 2000 |
| Phoneme | 20 | 20 | 1962 | 1961 |
| Pipeline Flow | 12 | 3 | 1000 | 1000 |
| Pima | 8 | 2 | 500 | 200 |

As a classifier we used Learning Vector Quantization (LVQ) using package LVQ_PAK [8]. One parameter of this classifier is the number of code vectors, that determine the decision borders between classes. These were chosen approximately according to the number of training examples for each database: Letter - 500, Landsat and Phoneme - 200, Pipeline Flow - 25, and Pima - 15 code vectors. Since the training procedure in LVQ is stochastic in nature - the end result exhibits minor variations as the presentation order of the training examples is varied - each accuracy figure presented is an average of ten LVQ classifiers trained with different random example presentation orders. Results comparing the linear transforms, Principal Component Analysis (PCA), LDA and MMI to nonlinear transforms are presented in Figures 2 - 6 in terms of classification accuracies on the test sets. Of these transforms, PCA is the only one that does not take the class labels of the training data into account.

Previous experiments showed that linear MMI-transforms can be significantly better than LDA or PCA transforms in capturing the essential information for classification [18]. Our experiments indicate that nonlinear transforms appear to be particularly useful in transformations to low dimensions, and especially in cases where the class borders have complex structures.

In view of these results, RBF networks appear to offer excellent capabilities as feature transforms to low dimensions, and in some cases, even to higher dimensions (Phoneme and Pima data sets).

Results using MLP networks are somewhat disappointing,

[1] See http://members.home.net/torkkola/mmi.html and compare the 2nd video clip (linear transform, $3d \rightarrow 2d$) to an RBF-transform starting from random parameters (5th video clip).

[2] http://www.cis.hut.fi/research/software.shtml

[3] http://www.ncrg.aston.ac.uk/GTM/3PhaseData.html

[4] http://www.ics.uci.edu/∼mlearn/MLRepository.html

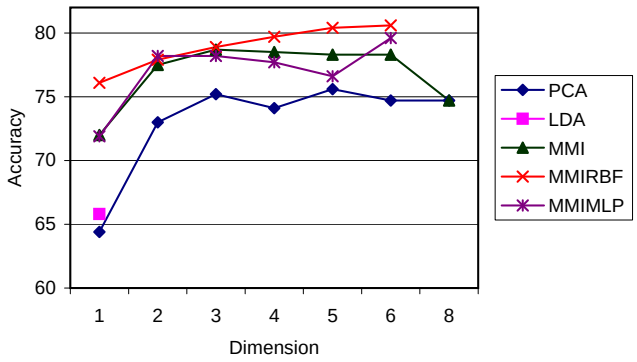


**Figure 2:** Accuracy on test data of the "Pima Indians Diabetes" data set using an LVQ classifier.

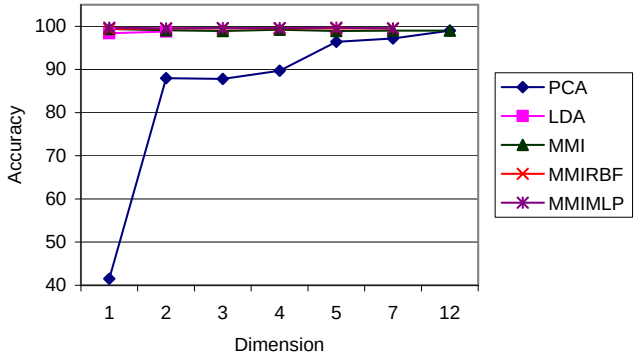


**Figure 3:** Accuracy on test data of the "Pipeline Flow" data set using an LVQ classifier.

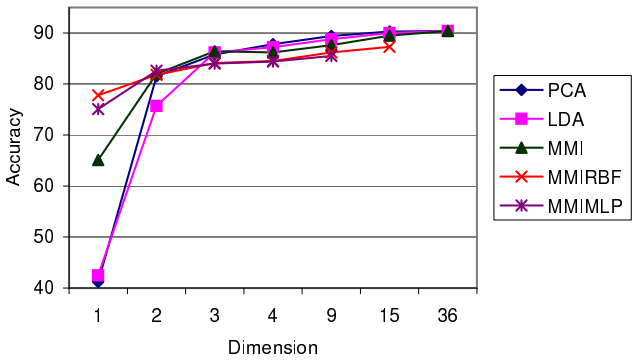


**Figure 4:** Accuracy on test data of the "Landsat Satellite Image" data set using an LVQ classifier.

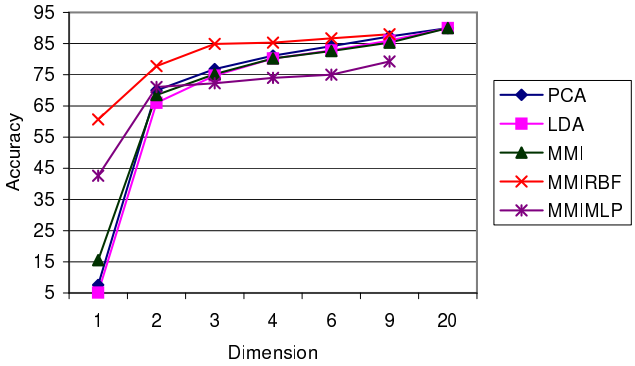


**Figure 5:** Accuracy on test data of the "Phoneme" data set using an LVQ classifier.

but those must be regarded as preliminary, since a more thorough exploration of the network structures and parameters was done using RBF networks.

A problem plaguing both nonlinear transforms, especially the MLP, appears to be the relative degradation of performance as the dimension of the output space increases. This could be attributable to one of a few reasons. First, Parzen density estimation does suffer from increasing dimensionality, espcially when the number of samples remains constant. This is the familiar "curse of dimensionality". There simply is not enough samples (regardless of the number) to reliably construct a kernel density estimate in higher dimensional spaces. A related issue is that the relative volume of the "sphere of influence" of a sample as determined by the kernel width, decreases exponentially as the dimension increases. This fact complicates the selection of an appropriate kernel width for a given output dimension.

Second, generalization may be an issue with flexible nonlinear transforms when coupled with small amounts of training data. A remedy would be enforcing "stiffer" transforms via regularization, for example, by using weight decay.

Third, local minima of the mutual information criterion are induced by the data itself, and a non-linear optimization technique might get stuck in these. A solution might be starting with a wide kernel, and decreasing the width during adaptation as done in some cases in [18]. A wide kernel might force larger clusters of data in different classes to separate first, and as the width is being shrunk, the adaptation would pay more and more attention to the finer details of the class distributions. Details of this scheme are left to future work.

Fourth, equation (9) has been derived on heuristic grounds, and to our knowledge, it lacks rigorous justification. Despite of this the measure appears to work well in practice.

A further caveat with the method lies in the nonparametric density estimation. The computational complexity of the method is $O(N^2)$, where $N$ is the number of samples. Basically the distances between every pair of samples in the output space have to be evaluated at each iteration. This might limit the applicability to huge data sets, but it is foreseeable that instance selection, clustering, or batch-based adaptation, where the square of the number of instances se-

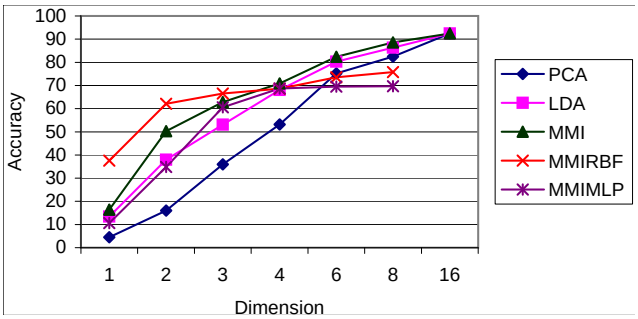


**Figure 6:** Accuracy on test data of the "Letter Recognition" data set using an LVQ classifier.

lected, clusters, or the batch size is set by computational limitations, could work well, too. Another possibility is to use semi-parametric density estimation methods.

## 7 Conclusion

This paper couples a nonparametric density estimator with a mutual information criterion based on Renyi's entropy to learn discriminative dimension-reducing transforms. Such transforms are useful in pattern recognition applications, where only the information that is essential to make classification decisions should be retained and the rest discarded. Resulting low-dimensional features enable a classifier to operate with less computational resources and memory without compromising the accuracy.

We learned nonlinear feature transforms implemented as neural networks, and compared the classification performance to linear transforms. Our experiments indicate that nonlinear transforms appear to be particularly useful in transformations to low dimensions, and especially in cases where the class borders have complex structures.

## References

[1] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *Neural Networks*, 5(4):537–550, July 1994.

[2] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, New York, 1995.

[3] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[4] R.M. Fano. *Transmission of Information: A Statistical theory of Communications*. Wiley, New York, 1961.

[5] J.W. Fisher III and J.C. Principe. A methodology for information theoretic feature extraction. In *Proc. of IEEE World Congress On Computational Intelligence*, pages 1712–1716, Anchorage, Alaska, May 4-9 1998.

[6] X. Guorong, C. Peiqi, and Wu Minhui. Bhattacharyya distance feature selection. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 2, pages 195 – 199. IEEE, 25-29 Aug. 1996.

[7] S. Haykin. *Neural Networks, A Comprehensive Foundation (2nd ed)*. IEEE press, New York, 1998.

[8] T. Kohonen, J. Kangas, J. Laaksonen, and K. Torkkola. LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms. In *Proceedings of the International Joint Conference on Neural Networks*, volume I, pages 725–730, Piscataway, NJ, 1992. IEEE.

[9] M.A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37:233–243, 1991.

[10] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, 1998.

[11] J. Mao and A.K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Trans. on Neural Networks*, 6(2):296–317, 1995.

[12] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A.J. Smola, and K.-R. Müller. Invariant feature extraction and classification in kernel spaces. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 526–532. MIT Press, 2000.

[13] T. Okada and S. Tomita. An optimal orthonormal system for discriminant analysis. *Pattern Recognition*, 18(2):139–144, 1985.

[14] J.C. Principe, J.W. Fisher III, and D. Xu. Information theoretic learning. In Simon Haykin, editor, *Unsupervised Adaptive Filtering*. Wiley, New York, NY, 2000.

[15] J.C. Principe, D. Xu, and J.W. Fisher III. Pose estimation in SAR using an information-theoretic criterion. In *Proc. SPIE98*, 1998.

[16] G. Saon and M. Padmanabhan. Minimum bayes error feature selection for continuous speech recognition. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 800–806. MIT Press, 2001.

[17] J. Sinkkonen and S. Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, (in press), 2001.

[18] K. Torkkola and W. Campbell. Mutual information in learning feature transformations. In *Proceedings of International Conference on Machine Learning*, Stanford, CA, USA, June 29 - July 2 2000.

[19] N. Vlassis, Y. Motomura, and B. Krose. Supervised dimension reduction of intrinsically low-dimensional data. *Neural Computation*, (in press), 2001.

[20] A.R. Webb. Nonlinear feature extraction ith radial basis functions using a weighted multidimensional scaling stress measure. In *Proc. 13th Int. Conf. on Pattern Recognition*, pages 635–639. IEEE, 25-29 Aug. 1996.

[21] H. Yang and J. Moody. Feature selection based on joint mutual information. In *Proceedings of International ICSC Symposium on Advances in Intelligent Data Analysis*, Rochester, New York, June 22-25 1999.