Feature Selection and Non-linear Feature Extraction

Shailesh Kumar
Department of Electrical and Computer Engineering
The University of Texas at Austin.

Abstract

Feature extraction and feature selection are two important tasks in pattern recognition. Classification algorithms like k-nearest neighbors, which are based on the assumption that patterns in the same class are close to each other and those in different classes are far apart (locality property), rely heavily on the quality of the features extracted from the input data. In this work, an objective function, which translates the locality property into a linear, Fisher like criteria, based on within and between class variance of the training data is proposed. This criteria is used for the two tasks of feature selection and feature extraction. Feature selection is done by introducing relevance measures for each input dimension. Feature extraction is defined as a linear combination of a set of non-linear functions. Closed form solutions for both, the relevance measures in feature selection and for the weights of linear combinations for feature extraction task are derived using the Fisher like criteria. Experiments over synthetic and real datasets have been used to highlight the strengths and weaknesses of these methods. Feature selection improves the performance of k-nearest neighbor classifier significantly for both synthetic and real data sets, while the feature extractor is found to be able to extract features from input spaces which are not suitable for simple classifiers like linear discriminant and k-nearest neighbors.

1 Introduction

Pattern recognition (Duda and Hart 1973; Bishop 1995; Fukunaga 1990) deals with assigning a class label $\omega \in \Omega = \{\omega_1, \omega_2, ..., \omega_c\}$ to a pattern vector $x \in R^d$. In parametric supervised learning framework, this is done by first *training* a pattern classifier $\phi(x, \Lambda) : R^d \rightarrow \Omega$, where Λ is a set of parameters in terms of which the mapping is learned. An emperical risk function $E(X, \Lambda)$ is minimized over a training set X of n labelled examples $\{(x^i,\omega^i)\}$, $i = 1 \ldots n$, using a learning algorithm. The set of parameters Λ^* for which $E(X, \Lambda)$ is minimum captures the distribution of the patterns in the pattern space among different classes. Mixture of Gaussians, Neural networks, Radial Basis functions, Mixture of Experts are some of the examples of parametric supervised learning methods (Bishop 1995). Each one of these methods have their own set of parametres to be learned, for example in mixture of gaussians, the means, variances and weights of all the gaussian kernels are the parameters while in a neural network the weights of the links and the biases are the parameters (Haykin 1994). Another class of supervised learning paradigm is *non-parametric* approaches where no parametric form of the calssifier is assumed and there are no parameters to train as such but, instead the data set X of labelled examples is used directly to make classification decisions for novel points. The generic form of a non-parmetric pattern classifier is $\psi(x, X)$. Nearest neighbour classifiers, Parzen windows (Duda and Hart 1973) etc fall into this category of pattern classifiers.

Essentially a pattern classifier tries to learn the distribution of the patterns in the pattern space among different classes, or in other words, it tries to learn how the pattern space is shared by different classes. The functions ψ for both parametric or non-parametric classifiers try to model these distributions. More often than not, however, the distribution of patterns in the patten space is not trivial. Non-parametric methods like nearest neighbour, parzen windows and paramteric methods like radial basis functions and mixture of experts rely heavily on the *locality property* (Fukunaga 1990) according to which:

Patterns that belong to the same class are close to each other and those in differnt classes are relatively farther away, according to some distance metric.

Patterns in the real world do not necessarily have the locality property. For example the vector of sensor readings from a chemical plant from which the state of the plant is to to be determined or a satellite image in which the goal is to identify an oil spill, or a pattern of diagnostic readings using which the health condition of a patient is to be determined, or for that matter, to identify a hand written character represented as a bit map, cannot be used directly for the purpose of classification. Some *features* must be extracted from these patterns before they can be classified. Thus there are two phases in pattern classification, first is feature extraction in which informative features are extracted from the input patterns and second is classification over these features. Let a function $f{:}R^d \rightarrow R^q$ transform the d-dimensional pattern space into a q-dimensional *feature space*. The generic form of the parametric and non-parametric classifiers now become $\psi(y(x), \Lambda){:}R^q \rightarrow \Omega$ and $\psi(y(x), X){:}R^q \rightarrow \Omega$. For methods that rely on the locality property, it is important that the features extracted by applying the transformation $y(x)$ are such that the locality property holds in the feature space. If the features extracted are good then the task of the classifier becomes easy. Hence feature extraction is a very important part of pattern classification.

Sometimes it might happen that there are a lot of possible features out of which only a few are useful. Using a large number of features leads to the curse of dimensionality and it is hard to find a good classifier which can generalize well if all these features are used at the same time. Hence there is a problem of *feature seclection* (Devijver and Schnabel 1982; Siedlecki and Sklansky 1988; Fukunaga 1990; Hand 1981) where only a subset of relevent features are to be selected to do the classification. For example the height of a person is not a good feature for disease diasgnosis and so on. Hence, it is important to find a measure of usefulness of a feature for a given problem. No direct methods of judging quality of a feature for the task of classification at hand are available. A subset of features is evaluated based on how well the set does on the task (Bishop 1995)

Although domain knowledge about the pattern/feature space can be used for feature selection and feature extraction, the knowledge may not be accurate or complete (Bishop 1995). Thus the challange in pattern classification is that, given the data, first, either using feature extraction i.e. transforming the input patterns into useful features, or using feature selection i.e. picking the best of the features, simplify the problem and then learn a classifier.

In this work, these two issues of feature selection and feature extraction have been addressed in light of the locality property. For feature selection a mesure of relevance of each feature is computed by minimizing an objective function. For feature extraction, parametric non-linear transformation of the pattern space into one-dimensional feature space are defined. The parameters of these transforms are then computed by optimizing a similar objective funciton used for feature selection.

2 Feature Selection

As mentioned above, some features might be more relavant than others to the problem at hand. Without using domain knowledge to handpick these relevant features, just using the training data for the job is of significance in domains where little is known about the input space. A feature vector $x = (x_1, x_2, ..., x_d)$ has d features. Let α_i, $i = 1 \ldots d$ be the relavance of the feature i to the problem at hand. Let X_ω be the set of training data points from class ω. Let $\delta(x, x')$ be the distance between two points x and x'. In this work we will use Eucledian distance given by:

$$\delta(x, x') = \sqrt{\sum_{i=1}^{d} (x_i - x'_i)^2} \tag{1}$$

Note that the distance metric assumes that all the dimensions are equally important. Nearest neighbor classifiers use the above distance metric (Duda and Hart 1973). Using the relevance measures, the above distance metric

can be modified to $\delta_\alpha(x, x')$ defined as follows:

$$\delta_\alpha(x, x') = \sqrt{\sum_{i=1}^{d} \alpha_i^2 (x_i - x_i')^2}, \tag{2}$$

with the constraint that $0 \leq \alpha_i \leq 1$ and that $\sum_{i=1}^{d} \alpha_i = 1$. Essentially equation ?? weighs each dimension according to its relevance to the problem. Parallels of this approach are the Mahalanobis distance which has a similar effect.

2.1 Problem Formulation

The problem now is to find the values of the d parameters α_i, given the data set. The goal is to find these relevance measures such that the locality property is satisfied. The approach taken in this work is to first define an objective function, which captures the essence of the locality property, in terms of the data set $X = \cup_\omega X_\omega$ and these relevance measures $\alpha_1 \ \ldots \ \alpha_d$, and then minimize this objective function to find the best relevance values. The objective function used in this work is based on the *within* and *between class* variance obtained directly from the training data.

Let μ^ω be the mean of class ω defined by:

$$\mu^\omega = \frac{1}{n_\omega} \sum_{x \in X_\omega} x, \tag{3}$$

where n_ω is the number of training data points in the set X_ω. Let μ be the mean of all data points given by:

$$\mu = \frac{1}{n} \sum_{x \in X} x = \frac{1}{n} \sum_{\omega \in \Omega} n_\omega \mu^\omega, \tag{4}$$

The **within class variance** $W(\alpha)$ is a measure of how close the points of a class are with each other. Essentially it measures the variance of data points within the same class. The within class variance is defined as:

$$W(\alpha) = \sum_{\omega \in \Omega} \sum_{x \in X_\omega} \delta_\alpha(\mu^\omega, x)^2 = \sum_{\omega \in \Omega} \sum_{x \in X_\omega} \sum_{i=1}^{d} \alpha_i^2 (\mu_i^\omega - x_i)^2. \tag{5}$$

The **between class variance** $B(\alpha)$ measures the separation of classes. It is defined in terms of the means of the classes and the mean of the whole data set as follows:

$$B(\alpha) = \sum_{\omega \in \Omega} n_\omega \delta_\alpha(\mu, \mu^\omega)^2 = \sum_{\omega \in \Omega} n_\omega \sum_{i=1}^{d} \alpha_i^2 (\mu_i - \mu_i^\omega)^2 \tag{6}$$

The locality property can be translated to an objective function $J(\alpha)$ in terms of the within and between class variance as follows:

$$J(\alpha) = B(\alpha) - \xi W(\alpha) - 2\lambda (\sum_{i=1}^{d} \alpha_i - 1) \tag{7}$$

ξ is used to weigh the within class verses the between class variance. 2λ is the Lagrangian multiplier for the constraint that the α_i's should sum up to 1. Maximizing $J(\alpha)$ translates to finding the relevance vector α for which the between class variance is as high as possible and the within class variance is as low as possible.

2.2 Optimal Relevance Measures

Once the criteria for optimization is defined, it is easy to find a closed form solution for the relevance measure by equating the d partial derivatives $\frac{\partial J}{\partial \alpha_k} = 0$, $k = 1 \ldots d$.

$$\frac{\partial J}{\partial \alpha_k} = \frac{\partial B(\alpha)}{\partial \alpha_k} - \xi \frac{\partial W(\alpha)}{\partial \alpha_k} - 2\lambda = 0 \tag{8}$$

$$\frac{1}{2}\frac{\partial J}{\partial \alpha_k} = \sum_{\omega \in \Omega} n_\omega \alpha_k (\mu_k - \mu_k^\omega)^2 - \xi \sum_{\omega \in \Omega} \sum_{x \in X_\omega} \alpha_k (\mu_k^\omega - x_k)^2 - \lambda = 0. \tag{9}$$

Solving for α_k we get:

$$\alpha_k = \lambda \left(\sum_{\omega \in \Omega} n_\omega (\mu_k - \mu_k^\omega)^2 - \xi \sum_{\omega \in \Omega} \sum_{x \in X_\omega} (\mu_k^\omega - x_k)^2 \right)^{-1} \tag{10}$$

The sufficient condition for which the solution given in equation (10) is a maxima on the optimality criteria is given by the constraint $\frac{\partial^2 J}{\partial \alpha_k^2} < 0 \ \forall \ k = 1 \ldots d$, which reduces the following constraint on the values of ξ:

$$\xi > \max_k \left(\frac{\sum_{\omega \in \Omega} n_\omega (\mu_k - \mu_k^\omega)^2}{\sum_{\omega \in \Omega} \sum_{x \in X_\omega} (\mu_k^\omega - x_k)^2} \right) \tag{11}$$

Once ξ is chosen in this range, it is easy to find λ, the normalizing factor as follows:

$$\lambda = \sum_{i=1}^{d} \left(\sum_{\omega \in \Omega} n_\omega (\mu_i - \mu_i^\omega)^2 - \xi \sum_{\omega \in \Omega} \sum_{x \in X_\omega} (\mu_i^\omega - x_i)^2 \right)^{-1} \tag{12}$$

Thus, just by looking at the training data, it is easy to find the optimal set of relevance parameters by maximizing an objective function which tries to impose the locality property on the feature space. A close examination of equation (10) reveals that essentially the relevance of a dimension evaluates to a value which is inversely proportional to the negative of the class separability in that dimension. In other words, if the class separability in certain direction is high then the relevance in that direction shall also be high. Experimental results discussed in section 4 show how this criteria works on synthetic and real data sets.

Another possible candidate for choosing relevance of a dimension based on the within and between class variance is ratio of the between class to within class variance. That is,

$$\alpha_k = \lambda \left(\frac{\sum_{\omega \in \Omega} n_\omega (\mu_k - \mu_k^\omega)^2}{\sum_{\omega \in \Omega} \sum_{x \in X_\omega} (\mu_k^\omega - x_k)^2} \right) \tag{13}$$

Where λ is the normalizing factor. This is based on the same intuition that for each dimension if it is useful in giving high separation between classes and high coherence within classes then its relevance is high. This criteria referred to as FISHER-RATIO is compared with the criteria derived above which is referred to as FISHER-LINEAR.

3 Non-linear Feature Extraction

As mentioned before, the task of feature extraction is to transform the pattern space into a feature space which helps to build a classifier by inducing the locality property in the feature space. In this work non-linear transforms that can be represented as a linear combinations of a family of non-linear basis functions have been considered. The objective function used to induce the locality property is the same as used above. The within and between class variance in the one dimensional feature space is expressed in terms of the coefficients of the linear combination and the objective function is maximized to find the optimal set of these coefficients.

3.1 Problem Formulation

A pattern space ($Xspace$) is transformed into a feature space ($Yspace$) using a non-linear transform $y : R^d \rightarrow R$ which transforms the d dimensional input in $Xspace$ into a one dimensional $Yspace$. Function $y(x)$ is defined as a linear combination of a set Φ of m non-linear functions $\{\phi_i(x)\}_{i=1}^m$ where each $\phi_i : R^d \rightarrow R$. The generic form of $y(x)$ is given by:

$$y(x) = \sum_{i=1}^{m} a_i \phi_i(x) = A^T \phi(x) \tag{14}$$

where $A = (a_1, a_2, ..., a_m)^T$ is the column vector of coefficients and $\phi(x) = (\phi_1(x), \phi_2(x), ..., \phi_m(x))^T$ is the column vector of non-linear functions in the set Φ.

Using the notation from previous section, the mean of class ω in the $Yspace$ μ^ω is defined as:

$$\mu^\omega = \frac{1}{n_\omega} \sum_{x \in X_\omega} y(x) = \frac{1}{n_\omega} \sum_{x \in X_\omega} A^T \phi(x) = A^T \left(\frac{1}{n_\omega} \sum_{x \in X_\omega} \phi(x) \right) = A^T \tilde{\phi}^\omega. \tag{15}$$

where $\tilde{\phi}^\omega$ is $m \times 1$ column vector given by:

$$\tilde{\phi}^\omega = \frac{1}{n_\omega} \sum_{x \in X_\omega} \phi(x). \tag{16}$$

Similarly the mean over all classes μ is defined as:

$$\mu = \frac{1}{n} \sum_{x \in X} y(x) = \frac{1}{n} \sum_{x \in X} A^T \phi(x) = A^T \left(\frac{1}{n} \sum_{x \in X} \phi(x) \right) = A^T \tilde{\phi}. \tag{17}$$

where $\tilde{\phi}$ is again an $m \times 1$ column vector given by:

$$\tilde{\phi} = \frac{1}{n} \sum_{x \in X} \phi(x) = \frac{1}{n} \sum_{\omega \in \Omega} n_\omega \tilde{\phi}^\omega \tag{18}$$

The **Between Class Variance** $B(A)$ is defined as:

$$B(A) = \sum_{\omega \in \Omega} n_\omega (\mu - \mu^\omega)^2 = \sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi} - A^T \tilde{\phi}^\omega)^2. \tag{19}$$

$$B(A) = (A^T \tilde{\phi})^2 \sum_{\omega \in \Omega} n_\omega + \sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi}^\omega)^2 - 2(A^T \tilde{\phi}) \sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi}^\omega). \tag{20}$$

Using equation (18) and the fact that $\sum_{\omega \in \Omega} n_\omega = n$,

$$\sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi}^\omega) = n (A^T \tilde{\phi}) \tag{21}$$

Substituting (21) in (20),

$$B(A) = n (A^T \tilde{\phi})^2 + \sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi}^\omega)^2 - 2n (A^T \tilde{\phi})(A^T \tilde{\phi}) \tag{22}$$

or

$$B(A) = \sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi}^\omega)^2 - n (A^T \tilde{\phi})^2 \tag{23}$$

The **Within class variance** $W(A)$ is defined as:

$$W(A) = \sum_{\omega \in \Omega} \sum_{x \in X_\omega} (\mu^\omega - y(x))^2 = \sum_{\omega \in \Omega} \sum_{x \in X_\omega} (A^T \tilde{\phi}^\omega - A^T \phi(x))^2 \tag{24}$$

$$W(A) = \sum_{\omega \in \Omega} (A^T \tilde{\phi}^\omega)^2 \sum_{x \in X_\omega} 1 + \sum_{\omega \in \Omega} \sum_{x \in X_\omega} (A^T \phi(x))^2 - 2 \sum_{\omega \in \Omega} (A^T \tilde{\phi}^\omega) \sum_{x \in X_\omega} A^T \phi(x) \tag{25}$$

Substituting

$$\sum_{x \in X_\omega} 1 = n_\omega \,, \tag{26}$$

$$\sum_{\omega \in \Omega} \sum_{x \in X_\omega} (A^T \phi(x))^2 = \sum_{x \in X} (A^T \phi(x))^2 \tag{27}$$

and

$$\sum_{x \in X_\omega} A^T \phi(x) = n_\omega (A^T \tilde{\phi}^\omega) \tag{28}$$

in equation (25) we get

$$W(A) = \sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi}^\omega)^2 + \sum_{x \in X} (A^T \phi(x))^2 - 2 \sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi}^\omega)^2 \tag{29}$$

or

$$W(A) = \sum_{x \in X} (A^T \phi(x))^2 - \sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi}^\omega)^2 \tag{30}$$

The criteria used to impose the locality property in the $Y space$ is given by:

$$J(A) = B(A) - \xi W(A) - 2\lambda (A^T I_m^* - 1) \tag{31}$$

where I_m^* is an $m \times 1$ column vector of ones that is $I_m^* = (1\ 1\ ...\ 1)^T$. The last term in equation (31) imposes a constraint on A such that:

$$A^T I_m^* = \sum_{i=1}^{m} a_i = 1. \tag{32}$$

ξ weights the within class variance with respect to the between class variance and 2λ is a Lagrangian multiplier for imposing the constraint given in equation (32).

3.2 The Optimal Solution

Minimization of $J(A)$ is done by setting $\frac{\partial J}{\partial A}$=0.

$$\frac{\partial J}{\partial A} = \frac{\partial B(A)}{\partial A} - \xi \frac{\partial W(A)}{\partial A} - 2\lambda I_m^* \tag{33}$$

(using identity: $\frac{\partial x^T P}{\partial x} = P$) (Zwillinger 1996)

$$\frac{\partial (A^T I_m^*)}{\partial A} = I_m^* \tag{34}$$

$$\frac{\partial B(A)}{\partial A} = 2 \sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi}^\omega) \tilde{\phi}^\omega - 2n (A^T \tilde{\phi}) \tilde{\phi} \tag{35}$$

and

$$\frac{\partial W(A)}{\partial A} = 2 \sum_{x \in X} (A^T \phi(x)) \phi(x) - 2 \sum_{\omega \in \Omega} n_\omega (A^T \tilde{\phi}^\omega) \tilde{\phi}^\omega \tag{36}$$

Substituting (35) and (36) in (33), we get:

$$\frac{1}{2}\frac{\partial J}{\partial A} = (1+\xi)\sum_{\omega\in\Omega} n_\omega (A^T\tilde{\phi}^\omega)\tilde{\phi}^\omega - n(A^T\tilde{\phi})\tilde{\phi} - \xi\sum_{x\in X}(A^T\phi(x))\phi(x) - \lambda I_m^* \tag{37}$$

Equating (37) to 0 and using the identity $(A^TB)B = (B^TB)A$ (Zwillinger 1996)

$$(1+\xi)\sum_{\omega\in\Omega} n_\omega (\tilde{\phi}^{\omega^T}\tilde{\phi}^\omega)A - n(\tilde{\phi}^T\tilde{\phi})A - \xi\sum_{x\in X}(\phi(x)^T\phi(x))A = \lambda I_m^* \tag{38}$$

or

$$\Theta A = \lambda I_m^* \tag{39}$$

where Θ is an $m \times m$ matrix:

$$\Theta = (1+\xi)\sum_{\omega\in\Omega} n_\omega (\tilde{\phi}^{\omega^T}\tilde{\phi}^\omega) - n(\tilde{\phi}^T\tilde{\phi}) - \xi\sum_{x\in X}(\phi(x)^T\phi(x)) \tag{40}$$

If Θ is non-singular, the optimal solution for A is given by:

$$A = \lambda\Theta^{-1} I_m^* \tag{41}$$

The condition that A indeed gives the maxima of the objective function in equation 31 is obtained by choosing ξ such that $\frac{\partial}{\partial a_k}\left(\frac{\partial J}{\partial A}\right) < 0$ In other words,

$$\xi > \max_{i,j=1\ldots m}\left(\frac{\sum_{\omega\in\Omega} n_\omega(\tilde{\phi}_i^\omega\tilde{\phi}_j^\omega) - n(\tilde{\phi}_i\tilde{\phi}_j)}{\sum_{x\in X}(\phi_i(x)\phi_j(x)) - \sum_{\omega\in\Omega} n_\omega(\tilde{\phi}_i^\omega\tilde{\phi}_j^\omega)}\right) \tag{42}$$

where i and j denote components of the various vectors.

4 Experiments and Results

Two sets of experiments, one for feature selection and other for feature extraction were performed. k-nearest neighbor classifier is used for evaluating the efficacy of both feature selection and feature extraction methods. The two sets of experiments are discussed below.

4.1 Feature Selection

In section 2, using a linear objective function to emulate the locality property, a criteria for computing relevances of various features given the training data was proposed. This criteria is based on the within and between class variance of the dataset. This criteria given in equation 31 is called the FISHER-LINEAR. Another criteria based on similar intuition, but using ratios instead of linear combination was propose, again based on the within and between class variance. It is called FISHER-RATIO (equation 13). These two feature selection criteria are compared with each other and the case where all features are equally weighed. The comparison was done on two datasets, one synthetic and the other the standard Iris data set from UCI repository.

The synthetic dataset used was 3 dimensional with 4 classes. The variance in the three dimensions were different. 40 data points in each class were generated. The Iris data set contains 3 classes. There are 50 points in each class. A feature vector is 4 dimensional.

For both these datasets, k-nearest neighbor classifier was used. For Synthetic data, k=5 was used for Iris data set, $k = 7$ was used. The results were generated as follows. In each test run, one of the n ($n = 160$ for synthetic and $n = 150$ for Iris) data points is chosen as the test pattern and the remaining $n - 1$ were used as training patterns. In case of FISHER -LINEAR and FISHER-RATIO, the relevance of the different dimensions were computed using the training patterns. The test sample was classified by three methods, (1) the regular k-nearest

Feature Selection Method	Synthetic Data Set	Iris Data Set
No Feature Selection	23.23 %	3.33 %
FISHER-RATIO	12.5 %	3.33 %
FISHER-LINEAR	7.5 %	2.00 %

Table 1: Error rates on synthetic and Iris data sets, The results for FISHE-LINEAR for the optimum choice of ξ values is given. For Synthetic data the value of 20 and for Iris data, the value of 10 was found to give the best results.

.

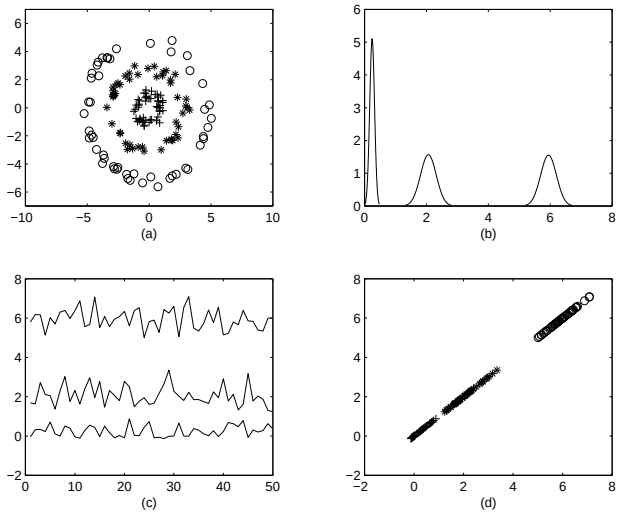

Figure 1: **Non Linear Feature Extraction**: (a) The data set distributed as concentric circles. Three classes with 50 data points each are shown. (b) The Gaussian distribution of the one dimensional feature space. (c) The actual values of the features $y(x)$ for the three classes. They are clearly distinguishable from each other. (d) the y values plotted on the line $y = x$.

neighbor classifier (with k as mentioned above) with no weights attached to different dimensions, (2) by using FISHER-LINEAR method of computing feature relevances and then doing k-nearest neighbor incorporating these relevance measures in the distance metric (with same values of k as in first set of experiments), (3) by using FISHER-RATIO method of computing feature relevances and using these in the k-nearest neighbor. The experiments were repeated n times, choosing each data point as the test sample in each run. Error rate in classification is given in table (1) below:

4.2 Feature Extraction

Features are extracted from the pattern space using non-linear transformations represented as linear combinations of non-linear functions. In this work, the feature space is one dimensional. A linear fisher like criteria which is minimized to induce the locality property in the feature space is used to extract these features. Two types of results are reported in this section. First the effect of the transformation of pattern space into one dimensional feature space is shown on a synthetic dataset which are not suitable for using Maximum likelihood Bayesian classifiers or simple linear discriminant functions. The second set of results deal with improvement in performance due to the feature extraction on a synthetic dataset and Iris data set.

Feature Selection Method	$\|\Phi\|$	k-Nearest Neighbor($k = 5$)
No Feature Extraction	NA	3.33 %
Feature Extraction (degree 2)	5	0.67 %
Feature Extraction (degree 3)	9	6.00 %

Table 2: Error rates on synthetic data set, using k-nearest neighbor. Feature extraction is done in two ways. In first only the first 5 terms of the set Φ which lead to degree two polynomials is used. In second all 9 terms of the set Φ which leads to degree 3 polynomials is used.

Feature Selection Method	$\|\Phi\|$	k-Nearest Neighbor($k = 7$)
No Feature Extraction	NA	3.33 %
Feature Extraction (degree 3)	10 (random)	8 %

Table 3: Error rates on Iris data set, using k-nearest neighbor.

Figure 1 shows the effect of transforming the 2 dimensional pattern space into a one dimensional feature space. The data points are distributed as three concentric circles with noise added to them. Points in the three circles belong to three different classes. Note that this distribution is not at all suitable for using Maximum Likelihood Bayesian classifier since the mean of all classes is the same and only the variance is different. Also, it would take a number of Gaussians to model the density of each class and hence it is feasible to transform this pattern space into a space which is more tractable with simpler classifiers. Using the non-linear transformation proposed in section 3, the one-dimensional feature space is given in figure 1(d). The family Φ used for this transformation comprises of all terms (except for a constant term) in a third degree multivariate polynomial in two variables (since input is two dimensional). The number of terms therefore is 9. If x_1 and x_2 are the two dimensions then the family Φ is given by:

$$\Phi = \{x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_2^3, x_1x_2^2, x_1^2x_2\} \tag{43}$$

Optimum values of the weighting coefficients were computed as proposed in section 3. The points in the feature space are used to fit three Gaussian distributions, one for each class. Figure 1(b) shows these Gaussian distributions fitted in the feature space.

In the set of results reported below, k-nearest neighbor is used in the pattern space and the feature space extracted as proposed in section 3. The error rates before and after feature extraction for both these classifiers are compared on two datasets, (1) the synthetic data set shown in figure 1(a) and (2) the Iris dataset from UCI repository. For k-nearest neighbor classifier, the procedure is the same as mentioned in section 3. For Results on the two datasets for two datasets are reported in tables 2 and 3.

Results on k-nearest neighbor show significant improvement in the synthetic data case while in the Iris dataset, the best performance after a number of random choices of the subsets of size 10 from the set Φ, was very poor compared to the benchmark performance of 3.33%. The problem with the feature extraction scheme proposed in section 3 is that it does not scale very well for high dimensional pattern space. For synthetic data, the dimensionality of the input space was just 2 and hence the exhaustive set of all possible functions Φ for generating degree 3 polynomials in 2 variables is just 9. For Iris dataset, on the other hand, the input is 4 dimensional and hence the exhaustive set of functions for generating degree 3 polynomials in four variables is 40. Performing linear combination on all 40 functions requires the inversion of a 40×40 matrix which in almost all cases is singular. As a result, a smaller subset of the set Φ is to be considered for dealing with this problem. In the results shown above, a number of random subsets of Φ of size 10 were tested and the results over the best such subset is reported.

5 Future Directions

The feature selection is a useful tool for high dimensional input spaces where the domain knowledge is not very useful in determining the relevant features. The approach proposed in this work can be extended in a number of ways by using different criteria for assigning relevance to each feature. The feature extraction method proposed in this work suffers from the drawback that it does not scale up too well with the dimensionality of the input space. In fact the number of possible functions increase exponentially with the dimensionality of input space. A search through a space of possible functions can be done effectively by using genetic algorithms. Moreover, domain knowledge can be used to choose proper families Φ for example if frequency data of a sound signal vector is a useful feature, Fourier function family or Wavelet function family can be used. If the input domain are images, Laplacian family of functions can be used and so on. Hence the proposed method provides a powerful framework of both intelligent search and use of domain knowledge to transform input space into tractable feature space.

6 Conclusion

Locality property in the input space, which is one of the basic assumptions of most of the classification methods, was used in this work for the tasks of feature selection and feature extraction. A Fisher discriminant like criteria, based on the within and between class variance is used as an objective function for obtaining relevance measures of different features in case of feature selection and for obtaining coefficients of the linear combination terms in case of non-linear feature extraction. Results over synthetic and real data set (Iris) show that feature selection does help in improving the performance of k-nearest neighbor. Feature extraction method is found to be useful in transforming input space, not suitable for simple linear discriminant functions to be used as classifier, into a feature space well suited for simple classification methods. On synthetic dataset, the k-nearest neighbor classifier showed significant improvement for low dimensional input data but as the dimensionality of the input space is more as in Iris data, the feature extraction method fails to scale up due to curse of dimensionality.

References

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Devijver, P. A., and Schnabel, R. B. (1982). *PAttern Recognition: A statistical Approach*. Englewoods Cliffs, NJ: Printice Hall.

Duda, R. O., and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. New York: Wiley.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. New York: Academic Press Inc.

Hand, D. J. (1981). *Discrimination and Classification*. New York: John Wiley.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. New York: Macmillan.

Siedlecki, W., and Sklansky, J. (1988). On automatic feature selection.

Zwillinger, D. (1996). *Standard Mathematical Tables and Formulae*. New York: CRC.