The Design
of
a Nonparametric Hierarchical Classifier

Chea-Tin (Tim) Tseng
and
Bernard M.E. Moret

Abstract

This paper proposes a method based on kernel density estimates to partition sequentially the feature space along the best feature axis (either one of the original axes or one obtained by a carefully developed one-dimensional linear feature transformation). This method alleviates the storage and classification speed problems of traditional kernel-based classifiers without losing their flexibility and their relative insensitivity to dimensionality.

We present a simple procedure and a distribution-free criterion to find a "good" smoothing parameter for the kernel density estimate. We develop a one-dimensional feature linear transformation based on correlation between density functions, which can be applied regardless of the geometrical structure of the data. Throughout our development, we validate our proposals by theoretical results and by simulations; we conclude with a sample application to real-world data (texture images).

1 Introduction

A classification procedure is a decision-making process which assigns an arbitrary pattern x of unknown class ω to any of c possible classes $\omega_1, \ldots, \omega_c$. The design of a classification procedure usually attempts to minimize the overall rate of classification error. Armed with a knowledge of the various probability distributions involved, we can compare the probabilities of x belonging to each class and classify according to maximum likelihood:

$$P(\omega_k|x) > P(\omega_j|x) \;\; \forall\, j \neq k \quad \Rightarrow \quad x \in \omega_k.$$

This fundamental rule is known as the *Bayes minimum error rule* and is always optimal (in probabilistic terms) with respect to the error rate. The *posterior* probability $P(\omega_i|x)$ can be obtained by applying Bayes' theorem:

$$P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{P(x)}$$

which yields

$$P(x|\omega_k)P(\omega_k) > P(x|\omega_j)P(\omega_j) \quad \Leftrightarrow \quad P(\omega_k|x) > P(\omega_j|x).$$

Thus, in theory, if the forms of the class-conditional probability density functions (pdf) $P(x|\omega_i)$ are known, the classification design problem can be solved by comparing the ratios of these functions with various thresholds. Since, in general, these forms are not known, parametric or nonparametric methods are used to estimate them. Parametric methods assume that the conditional probability density belongs to a certain family of functions (for example, normal distributions) and attempts to derive values for the parameters characterizing an individual distribution; however, in many pattern recognition problems, this assumption may not be justifiable (especially when the true probability is multimodal).

In contrast to parametric methods, nonparametric methods make no assumption regarding the underlying probability distribution [3,4,16]. Two major methods of nonparametric pdf estimation exist. One is the histogram density estimate; its main disadvantage is that the resulting estimate density either is discontinuous or does not accurately reflect the local variations in density. The other method is the kernel methods [2,8,10]. The kernel method estimates the pdf $\hat{f}(x)$ by selecting a suitable kernel function $K(x)$ to convolve with the empirical cumulative distribution function F_n:

$$\hat{f}(x) = \int K(x-y)dF_n(y)$$

Estimation by kernel methods have several advantages; they are: (i) consistent with Bayes' rule; (ii) flexible; and (iii) less sensitive to dimensionality than other methods [15]. However, because of their computation and storage costs, the kernel methods have yet to be widely applied to practical recognitions problems.

In view of the difficulties involved in the estimation of probability density functions, many other classification design methods have been proposed which do not rely on knowledge of pdf forms. One of these, the space partition method, has been the object of much attention by the pattern recognition community, as it has several advantages. First, it has a natural decision tree structure and therefore inherits the advantages of binary decision, namely: (i) the complex global decision making procedure can be divided into a number of simpler and local decision stages; (ii) at each stage, the feature subset which is best suited for the classification task can be selected; and (iii) the classification process is efficient. Secondly, the design procedure and resulting system is simple. Finally, the method does not depend on statistical distributions.

The space partition design method is based on the concept of "statistically equivalent blocks," introduced by Anderson [1] in 1966. The basic goal is to partition the observation space for each pattern class into regions which have similar statistical properties. Many researchers have successfully used this concept with different partition techniques [5,6,7,9,11,12,13]. Existing classification methods based on space partitioning all have one or more significant drawbacks: most suffer from boundary determination problems; some suffer from dimensionality problems; several are restricted to two-class cases; all require estimation (or *a priori* knowledge) of several parameters which cannot, in general, be known or determined with sufficient accuracy. In order to take advantage of the space and time efficiency of space partition methods, then, we must combine them with some robust kernel estimation procedure.

This paper is organized in six sections. In section 2 we propose a nonparametric classification phase 1 algorithm. And we refine our algorithm with smoothing parameter choosing and feature transformation in section 3, 4 respectively. In this section, The results of primary simulations—without using any feature transformation and without the benefit of our proposed method for choosing a smoothing parameter was presented. In section 3, a simple location-dependent procedure for the choice of a smoothing parameter and a distribution-free goodness-of-fit were developed. In section 4, we use an expansion through Hermite functions to estimate the correlation between density functions, which enables us to develop a good linear one-dimensional feature extraction procedure (for use in space partitioning techniques). We prove that, with the proper choice for the truncation order of the expansion, our estimate converges to the true density. section 5 discusses a real world application. We apply our proposed classification scheme with our proposed estimation and selection techniques to texture images and compare our results with those obtained with a k-NN classifier (in two versions, one using normalized and the other unnormalized features).

This application to a severely undersampled problem shows that our classifier is well suited to real-world applications, as it is insensitive to scaling and only slightly sensitive to dimensionality, two problems which plague most existing classifiers. The summary and conclusion were in the last section.

2 A Nonparametric Classification Algorithm and Its Primary Simulation

In the previous section, we described several advantages of kernel-based classifiers. Unfortunately, like nearest neighbor classifiers, kernel-based classifiers require the storage of a potentially large design set and suffer from fairly low classification speed. This can be a critical disadvantage in a system where the speed of classification is an important factor and where some accuracy can be sacrificed (for example, when a very large number objects must be classified very quickly in a satellite picture).

The storage and classification speed problems of kernel-based classifiers can be eased if the design samples can be replaced by a few sample points (as is done for nearest neighbor classifiers through editing and condensing procedures). Hence I propose a method based on the marginal kernel density estimate and on the idea of hierarchical classification and/or space partition. The proposed method would partition each feature axis (or subspace) into intervals (or regions) so that only few boundary points need be stored and a new pattern can be sequentially classified very quickly.

2.1 Combining Marginal Kernel Density Estimates and Space Partitions

We use the Bayes rule with a kernel density estimate to partition the observation space into successively finer regions for classification purposes. The new patterns can be sequentially classified within the resulting regions. For ease of explanation, the procedure will be described separately for one-dimensional and multi-dimensional cases.

One-dimensional classification Let $X_i = \{x_{i,1}, \ldots, x_{i,n_i}\}$ be a set of n_i independent design samples from class ω_i, for $i = 1, ..c$. Let p_i, which is usually estimated by $\frac{n_i}{\sum_{j=1}^{c} n_j}$, be the prior probability of class ω_i. Let $\hat{f}_{i,n_i}(x)$ be the kernel density estimate for $f_i(x)$, this last being the true density of class ω_i; and let $\hat{f}_{n_i}(\omega_i|x)$ be the posterior conditional probability estimate induced by the kernel density estimate. Furthermore, let ω_0 be the equivocation class, $\lambda(\omega_i|\omega_j)$ be the loss due to misclassifying samples from class ω_j in ω_i, and λ_r the loss for equivocation.

Phase one of the algorithm consists of four steps:

1. Order the samples in design set $X = X_1 \cup X_2 \cup \ldots \cup X_c$ in ascending fashion, say $\{y_1, \ldots, y_n\}$ $n = \sum_{j=1}^{c} n_j$.

2. Assign class label i to y_p for $i \in \{0, \ldots, c\}$, $p \in \{1, \ldots, n\}$ and form a class label array (y_p, i) by assigning y_p to class ω_0 (equivocating), if

$$\sum_{j=1}^{c} \lambda(\omega_i|\omega_j)\hat{f}_{ni}(\omega_i|y_p) > \lambda_r$$

   and by assigning y_p to class ω_i if

$$\sum_{j=1}^{c} \lambda(\omega_i|\omega_j)\hat{f}_{ni}(\omega_i|y_p) =$$

$$\min_{k=0}^{c} \sum_{j=1}^{c} \lambda(\omega_k|\omega_j)\hat{f}_{ni}(\omega_i|y_p)$$

   and

$$\sum_{j=1}^{c} \lambda(\omega_i|\omega_j)\hat{f}_{ni}(\omega_i|y_p) < \lambda_r$$

3. Adjust the boundaries by checking the class label for each y_i sequentially:

   (a) if y_i and y_{i+1} have the same class label, say ω_j, then delete (y_i, j) from class label array;

   (b) if y_i and y_{i+1} have different class labels, say ω_j and ω_k respectively, then use the bisection method to find a $y \in (y_i, y_{i+1})$ such that $\hat{f}_{j,n_j}(\omega_j|y) \approx \hat{f}_{k,n_k}(\omega_k|y)$ and replace y_i by y in the class label array.

4. Find the total loss (the sum of the misclassification loss and of the equivocation loss) for the partition.

Multi-dimensional classification For the multi-dimensional case, we apply the above one-dimensional algorithm to each feature axis or optimal one-dimensional transformed feature based on available samples. The feature associated with the smallest total loss is selected as dominant feature. The procedure is repeated on all equivocated design samples in the dominant feature until the small total loss is greater than preset threshold or the number of design samples falls below a preset number. To verify our phase one algorithm, we ran some simulations to compare performance our classifier with kernel-based classifiers; these are reported in the next section.

2.2 Primary simulation results

All following primary simulations concentrate on comparing classification performance between kernel density estimates with all features (the traditional kernel-based method) and our proposed classification procedure. The first two simulations compare classification performance when the features are highly correlated.

The first simulation uses 300 points for each class, selected independently from multivariate normal distributions, $\mathcal{N}(\underline{\mu}_1, \Sigma)$ for class 1, and $\mathcal{N}(\underline{\mu}_2, \Sigma)$ for class 2. The parameters are set as follows:

$$\mu_1 = [0\ldots, 0]_{1\times 6}$$

$$\mu_1 = [0.5, \ldots, 0.5]_{1\times 6}$$

$$\Sigma = \begin{bmatrix} 1 & 0.7 & \ldots & 0.7 \\ 0.7 & 1 & \ldots & 0.7 \\ \vdots & \vdots & \vdots & \vdots \\ 0.7 & 0.7 & \ldots & 1 \end{bmatrix}_{6\times 6}$$

We choose 150 points from each class to design the classifier and 150 points as test samples. The Cauchy kernel $\frac{1}{\pi(1+x^2)}$ and smoothing parameter $h = n^{-\frac{2}{5}}$ (which was suggested by K. Fukunaga) are used in the kernel density estimate. The results with our method are given below.

class	1	2
1	96	54
2	33	117

Only features 1, 5, and 2 were used in the classification procedure. Using the traditional kernel density estimate (i.e., with all features), we obtain:

class	1	2
1	55	95
2	31	119

The second simulation is similar to the first; the only change is in the means, where $\underline{\mu}_2$, from $[0.5, \ldots, 0.5]$, becomes $[1.5, \ldots, 1.5]$. The results with our method are given below.

class	1	2	0
1	105	44	1
2	7	143	0

Only 2 features were used in the classification procedure. Using the traditional kernel density estimate, we get:

class	1	2
1	102	48
2	12	138

From these two simulations, we see that with high feature correlation and heavy class overlap, our proposed classifier outperforms the classical classifier derived from kernel density estimates. With well-separated classes, both classifiers have approximately the same classification performance.

The iris data, which have four highly correlated features and three fairly well separated classes, was published by Fisher in 1936. The sepal length, sepal width, petal length, and petal width were measured in millimeters on 50 iris specimens from each of three species, *I. setosa*, *I. versicolor*, and *I. virginica*. We pick independently 25 points from each class to be design samples and use the remaining data as test samples. The results with our method are given below.

class	1	2	3
1	25	0	0
2	0	24	1
3	1	2	22

Only two features (3 an 4) were used in the classification procedure. Using traditional kernel density estimate, we get:

class	1	2	3
1	25	0	0
2	0	25	0
3	0	2	23

The results are comparable, but all four features were used.

The next two experiments compare the classification performance of the two classifiers when the features are uncorrelated and test the sensitivity of design data size *versus* data dimensionality.

Our third simulation is similar to our first, with one change in the standard deviation: $\Sigma = I_{6\times 6}$ The results with our method are given below.

class	1	2
1	95	55
2	18	132

Only features 3, 1, and 5 were used in the classification procedure. Using the traditional kernel density estimate we get:

class	1	2
1	85	65
2	20	130

Our fourth simulation is similar to our third, but we pick independently 25 points from each class to be design samples, and 25 points from each class to be test samples. The results with our method are given below.

class	1	2	0
1	18	6	1
2	9	16	0

Only one feature was used in the classification procedure. Using the traditional kernel density estimate we get:

class	1	2
1	11	14
2	1	24

The results of these last two simulations show that, when the features are uncorrelated and the sample design size is large, both classifiers have approximately the same classification performance. But the traditional kernel-based classifier used all available features, whereas our proposed method only used one feature, so that our proposed method is likely to be less sensitive to dimensionality.

All in all, our primary simulations results are encouraging, as they appear to indicate that our proposed method is preferable to the traditional kernel method. However, to be a more powerful and broad applicable classifier, a delicate and simple smoothing parameter and feature transformation need to be developed.

3 The choice of smoothing parameter

In step 2 of phase 1 algorithm, we will use the kernel method to estimate probability density function. The estimation procedure uses a kernel function $K(x)$ which itself is a density function; (in practice, a common choice of $K(x)$ is the normal density). Having selected $K(x)$, then the estimate of density function $f(x)$, using independent samples $x_1, x_2, ..., x_n$ from $f(x)$, is given by

$$\hat{f}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_n^n} K(\frac{x - x_i}{h_n})$$

where h_n is a smoothing parameter. Usually, the smoothing parameter is chosen so as to optimize some criterion in the asymptotic sense; this method is called location-independent.

The effect of the smoothing parameter h_n on the estimate $f_n(x)$ can be briefly analyzed as follows. If h_n is very large, the function $K(\frac{x}{h_n})$ becomes a broad, slowly changing function, and the resulting estimate $\hat{f}_n(x)$ is an over-smooth estimate of $f(x)$. On the other hand, for small h_n, $\hat{f}_n(x)$ is a spiky estimate of $f(x)$. In either case we are unable properly to replace $f(x)$ by $\hat{f}_n(x)$ in a classification procedure. Therefore, the choice of the smoothing parameter h_n is critical for density estimation based on kernels.

Although the Parzen kernel density estimate chooses an asymptotically optimal h_n for all samples $\{x_1, x_2, \ldots, x_n\}$, the situation is far from satisfactory in the finite sample case, as the estimate will be over-smoothed in regions of high variation and under-smoothed in regions of low variation. To counter this effect some authors have proposed to choose some h' by considering the local density of sample points. Usually the local density is defined in terms of distance between a point and its kth nearest neighbor in the sample set. This is the essence of the location-dependent method.

Breiman *et al.* [2] seem to have been the first to propose a method to obtain h with the kth nearest neighbor procedure in practical application.

For a sample point x_j they choose h_j to be

$$h_j = \alpha_k d_{j,k},$$

where α_k is a constant multiplier and $d_{j,k}$ is the distance from x_j to its kth nearest neighbor (omitting x_j itself) in the design set. They use a goodness-of-fit criterion to determine k and α_k, the two parameters needed to obtain each h_j. This goodness-of-fit is based on the fact that if $f(x)$ is the true density, then the variables

$$w_j = e^{-nf(x_j)V(d_{j,1})}, \quad j = 1, \ldots, n,$$

where $V(r)$ is the volume of an m-dimensional sphere of radius r (assume that each pattern vector has m components), have a univariate distribution that is approximately uniform. Thus the test statistic for an estimate $\hat{f}(x)$ is based on the variables

$$\hat{w}_j = e^{-n\hat{f}(x_j)V(d_{j,1})}, \quad j = 1, \ldots, n.$$

Now pick k and α_k which minimize the test statistic

$$\hat{S} = \sum_{j=1}^{n}(\hat{w}_{(j)} - \frac{j}{n})^2$$

where $\hat{w}'_{(j)}$ is the order statistic of the variables $\hat{w}'_j$. Breiman *et al.* ran simulations for unimodal and mixed mode normal distributions and found that the best location-dependent estimator was always superior to the best location-independent one. They also point out that the choice of a smoothing parameter using their method was much more sensitive to the selected criterion than the choice using Parzen's method. Unfortunately, choosing an optimal k for the kth nearest neighbor method and its corresponding constant α_k is very time-consuming. For simplify our classification design procedure, a new location-dependent procedure which is computationally simpler than Breiman's, yet produces equally good results, needed.

3.1 Smoothing Parameter Choosing Method

A good kernel density estimate must adapt properly to the local sample density. In regions of low sample density, the kernel will be spread out and the smoothing parameter h will be large. In high density regions, the converse will occur. In other words, the smoothing parameter h_k for the kernel centered at x_k is sensitive to the density of x_k's neighbors and relatively insensitive to densities in regions are far away from x_k. Based on this idea, we proposed [15] to set the smoothing parameter h_k as follows:

$$h_k = \frac{\sum_{j=1}^{n} d_{j,k}/d_{j,k}^p}{\sum_{j=1}^{n} 1/d_{j,k}^p}$$

$$= \frac{\sum_{j=1}^{n} 1/d_{j,k}^{p-1}}{\sum_{j=1}^{n} 1/d_{j,k}^p},$$

where n is the number of design samples, $d_{j,k}$ is the distance from x_k to its jth nearest neighbor. The first author had shown [14,15] that the density estimate using $h_k^{'}s$ is Bayes risk consistent. p is a number that can be determined by a goodness-of-fit statistic S_n.

The goodness-of-fit $S_n^{'}$ is based on that fact if $\hat{f}_n(x)$ is a density estimate of true density $f(x)$ based on n ordered sample $x_(1), x_(2), ..., x_(n)$, then if the statistic

$$S_n^{'} = \sum_{i=0}^{n} |\hat{F}_n(x_{(i+1)}) - \hat{F}_n(x_{(i)}) - \frac{1}{n+1}| + D_n$$

converge to $\frac{2}{e}$, the $\hat{F}_n(x)$ converge to $F(x)$ [15]. Where $\hat{F}_n(x)$, $F(x)$ are distribution of $\hat{f}_n(x)$ and $f(x)$ respectively.

Since the kernel density estimate $\hat{f}_n(x)$ can be represented by

$$\hat{f}_n(x) = \frac{1}{n}\sum_{j=1}^{n} K(\frac{x-x_j}{h_j}),$$

the quantile intervals can be computed by

$$\hat{F}_n(x_{(i+1)}) - \hat{F}_n(x_{(i)}) = \frac{1}{n}\sum_{j=1}^{n}\int_{x_{(i)}}^{x_{(i+1)}} K(\frac{x-x_j}{h_j})dx.$$

Thus, once the kernel function K was chosen, there is no difficulty in applying $S_n^{'}$ to kernel density estimates. It had been shown that

$$\lim_{n\to\infty} S_n^{'} = \sum_{i=0}^{n} |\hat{F}_n(x_{(i+1)}) - \hat{F}_n(x_{(i)}) - \frac{1}{n+1}| \leq \frac{2}{e}$$

and if

$$\lim_{n\to\infty} S_n^{'} = \sum_{i=0}^{n} |\hat{F}_n(x_{(i+1)}) - \hat{F}_n(x_{(i)}) - \frac{1}{n+1}| = \frac{2}{e}$$

then $\hat{F}_n(x) \rightarrow F(x)$ as $n \rightarrow \infty$.

Therefore, a natural choice is to select a good smoothing parameter by minimizing $S_n^{'}$. It is worth noting that, if the minimum value of $S_n^{'}$ is greater than some critical value, this indicates that the method is not a suitable density estimator for finite samples. Minimizing

$$S_n^{'} = \sum_{i=0}^{n} |\hat{F}_n(x_{(i+1)}) - \hat{F}_n(x_{(i)}) - \frac{1}{n+1}| + D_n$$

is obviously equivalent to minimizing each quantile interval difference $|\hat{F}_n(x_{(i+1)}) - \hat{F}_n(x_{(i)}) - 1/(n+1)|$.

However, consider the case of a monotonic decreasing kernel K, with $K(x) \geq 0$ for all x and $h_j \rightarrow 0$ for each $j = 1, \ldots, n$. Then

$$\hat{F}_n(x_{(i+1)}) - \hat{F}_n(x_{(i)}) = \frac{1}{n} \sum_{j=1}^{n} \int_{x_{(i)}}^{x_{(i+1)}} K(\frac{x - x_j}{h_j}) dx \rightarrow \frac{1}{n}.$$

This also implies $D_n \rightarrow 0$ and $S_n \rightarrow 2/(n+1) + (n-1)/(n(n+1)) \approx 3/(n+1)$. Consequently, the minima of $S_n^{'}$ will occur at $h_j = 0$ for each $j = 1, \ldots, n$. This is far from our purpose, as the estimate f_n will then become a series of probability spikes at sample points x_j, $j = 1, \ldots, n$ (and will be identically zero elsewhere).

The difficulty arises because of the values of $K(\frac{x_{(i)} - x_j}{h_j})$ or $K(\frac{x_{(i+1)} - x_j}{h_j})$ when $x_{(i)}$ or $x_{(i+1)}$ is very close to x_j. To overcome this difficulty, we replace the quantile interval by

$$\hat{F}_n(x_{(i+1)}) - \hat{F}_n(x_{(i)}) = \frac{1}{n - |N_{i,k}|} \sum_{x_j \notin N_{i,k}} \int_{x_{(i)}}^{x_{(i+1)}} K(\frac{x - x_j}{h_j}) dx$$

where $N_{i,k} = \{x_p |\ |x_p - x_{(i)}| < d_{i,k},\ 1 \leq p \leq n\}$ and $d_{i,k}$ is the distance between $x_{(i)}$ and its kth nearest neighborhood. Experimentation indicates that a good choice for k is $k = [\sqrt{n}]$ when kernel $K(x)$ is a normal density function.

The authors [15] ran extensive simulations to compare this method to Breiman's method. It showed that its performance with the MPE criterion is comparable to Breiman's method at a much low computational cost.

4 One Dimensional Linear Feature Transformation

In order to reduce dimensionality problems and thus indirectly enhance class separation, pattern recognition design often seeks a transformation H which maps an n-dimensional feature space into a lower, m-dimensional space. Since we intend to use a space partitioning method, we seek a linear transformation H which maps the feature space into a one-dimensional space with "good" class separation. In other words, with some separability criterion Q, we wish to find a unit vector $\underline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ such that criterion Q is optimized (or nearly so) along axis $\underline{\alpha}$ on the available samples.

Generally speaking, there are two major categories of class separability criteria. The first is based on interset and intraset distances; the idea of these criteria is to minimize the internal scatter of each (transformed) class while maximizing the distance between (transformed) samples of different classes. The advantage of such criteria is that optimal (or suboptimal) linear transformations are easily found through eigenvalue analysis. Drawbacks are directly related to the very assumptions implicit in the use of scatter matrices: since only means and deviations are taken into account, the distributions are effectively assumed to be normal, so that, if one or more classes follow multimodal distributions or consist of a sheet of sample points, eigenvalue analysis will generally not result in a transformation that achieves "good" class separation. The other category of separation criteria is based on the overlap of the class probability densities. Such criteria do not depend on the geometrical structure of the samples; however, since they require probability densities in the transformed space, optimization becomes unfeasible.

We will use criterion based on the correlation of probability densities:

$$Q = \begin{cases} \dfrac{\int p_1 p_2 f_1(x) f_2(x) dx}{\prod_{i=1}^{2} \sqrt{\int f_i^2(x) dx}} & \text{for two class case} \\ \sum_{i \neq j} \dfrac{\int p_i p_j f_i(x) f_j(x) dx}{\prod_{m=i,j} \sqrt{\int f_m^2(x) dx}} & \text{for } N\text{-classes case} \end{cases}$$

where the p_is are the prior probabilities and the $f_i(x)$ class conditional probabilities. We intend to use criterion Q to find an unit vector $\underline{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n)$ such that samples in R^n projected along this direction demonstrate "good" class separability. This is a constrained optimization problem:

$$minimize objective function: Q(\underline{\alpha}) = \sum_{i\neq j} p_i p_j \frac{\int f_i(z) f_j(z) dz}{\prod_{m=i,j} \sqrt{\int f_m^2(z) dz}}, \quad z = \underline{\alpha}^t . \underline{x}, subject to |\underline{\alpha}| = 1$$

Using normalized Hermite orthogonal functions expansion, the quantity Q can be totally determined by the expectation values of Hermite functions, which in turn can be easily estimated by the average of Hermite functions on samples [16]. The difficulty of density estimation in transformed space will be reduced, thereby making optimization feasible.

It is known that the normalized Hermite functions are

$$P_k(x) = \frac{e^{-\frac{x^2}{2}} H_k(x)}{\sqrt{2^k k! \sqrt{\pi}}} \quad k = 0, 1, 2, \ldots,$$

described by the recursive relation:

$$\begin{cases} P_0(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt[4]{\pi}} \\ P_1(x) = \frac{e^{-\frac{x^2}{2}} 2x}{\sqrt{2\sqrt{\pi}}} \\ P_k(x) = \sqrt{\frac{2}{k}} x P_{k-1}(x) - \sqrt{\frac{k-1}{k}} P_{k-2}(x) \text{ for } k \geq 2 \end{cases}$$

Thus, provided that $f_i(x) \in L_2$ (i.e., $f_i(x)$ is square integrable), $f_i(x)$ can be expanded in the orthogonal series

$$f_i(x) = \sum_{j=0}^{\infty} a_{j,i} P_j(x)$$

with the coefficients defined by

$$a_{j,i} = \int f_i(z) P_j(z) dz = E_i[P_j(z)]$$

so that

$$\begin{aligned} &= \int_{-\infty}^{\infty} f_i(z) f_j(z) dz \\ &= \int_{-\infty}^{\infty} \sum_{m=0}^{\infty} a_{m,i} P_m(z) f_j(z) dz \\ &= \sum_{m=0}^{\infty} a_{m,i} \int_{-\infty}^{\infty} P_m(z) f_j(z) dz \\ &= \sum_{m=0}^{\infty} a_{m,i} a_{m,j} \end{aligned}$$

and Q can also be rewritten as

$$Q = \begin{cases} \frac{\sum_{m=0}^{\infty} a_{m,1}a_{m,2}}{\prod_{i=1}^{2}\sqrt{\sum_{m=0}^{\infty} a_{m,i}^2}} & \text{for two class case} \\ \sum_{i\neq j} \frac{\sum_{m=0}^{\infty} p_i p_j a_{m,i}a_{m,j}}{\prod_{q=i,j}\sqrt{\sum_{m=0}^{\infty} a_{m,q}^2}} & \text{for } N\text{-classes case} \end{cases}$$

In geometrical terms, the Q indicates the angle formed by the vectors $f_1(x)$ and $f_2(x)$ for two classes case. When $f_1(x)$ and $f_2(x)$ are perpendicular to each other, the two classes have perfect separability in direction $\underline{\alpha}$; conversely, when $f_1(x) = f_2(x)$, we are totally unable to distinguish between the two classes.

Computing the criterion Q The infinite sum $Q(\underline{\alpha}) = \sum_{j=0}^{\infty} a_{j,1}a_{j,2}$ must be truncated at some order J for computational purposes, yielding the estimate

$$Q_J(\underline{\alpha}) = \sum_{i\neq j} p_i p_j \frac{\sum_{m=0}^{J} a_{m,i}a_{m,j}}{\prod_{q=i,j}\sum_{m=0}^{J} a_{m,q}^2}$$

In practical applications, the computation of $Q(\underline{\alpha})$ is based on data samples and the coefficients $a_{j,i}$—the expectations of normalized Hermite functions—are estimated by the corresponding sample average

$$\begin{aligned} \hat{a}_{j,i} &= \frac{1}{n_i}\sum_{k=1}^{n_i} P_j(z_{k,i}) \\ &= \frac{1}{n_i}\sum_{k=1}^{n_i} P_j(\underline{\alpha}^t x_{k,i}). \end{aligned}$$

and the prior probabilities $p_i's$ can be estimated by class sample ratio $\frac{n_i}{\sum_q n_q}$. Replacing $a_{j,i}$ by $\hat{a}_{j,i}$ in the truncated version Q_J, we obtain

$$\hat{Q}_J(\underline{\alpha}) = \sum_{i\neq j} \frac{n_i n_j}{\sum_q (n_q)^2} \frac{\sum_{m=0}^{J}(\sum_{k=1}^{n_i} P_m(z_{k,i}))(\sum_{k=1}^{n_j} P_m(z_{k,j}))}{\prod_{r=i,j}\sqrt{\sum_{m=0}^{J}(\sum_{k=1}^{n_r} P_m(z_{k,q}))^2}}$$

The truncated term J can set to be $\sqrt{\frac{n_1+n_2}{2}}$ [16], in practice.

To find an unit vector $\underline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ along which projected samples have "good" class separability, we can minimize $\hat{Q}_J$ with respect to the n components of $\underline{\alpha}$ under the constraint $\underline{\alpha\alpha^t} = 1$. To this end we use the method of *Lagrange* multipliers; this results in the system of equations $G(\underline{u}) = \underline{0}$, where

$$G(\underline{u}) = \begin{pmatrix} (\nabla_{\underline{\alpha}}(\hat{Q}_J(\underline{\alpha}) + \lambda(\underline{\alpha\alpha^t} - 1)))^t \\ \nabla_{\lambda}(\hat{Q}_J(\underline{\alpha}) + \lambda(\underline{\alpha\alpha^t} - 1)) \end{pmatrix}$$

where $\underline{u} = (\alpha_1, \alpha_2, \ldots, \alpha_n, \lambda)$. Any nonlinear root finding, such as *Newton-Raphson* algorithm, can be applied to find the zeros of $G(\underline{u})$; i.e. the optimal unit vector $\underline{\alpha}$.

5 Application in Image Classification

In this section, we present an application of our proposed nonparametric algorithm to the classification of texture images. Textures will include water, wool, fur, fiber, cork, grass, and similar objects.

Since Bela Julesz did some visual experiments which indicated that textures are important features for human perception, automatic texture classification has been an active area of research. Texture features have been successfully used in medical and remote-sensing imagery classification, while current research attempts to use them in difficult but important areas such as image segmentation, surface orientation, and object depth inference.

Our main concern in this study is to test the performance of our nonparametric classifier design on real data applications, not to study textures themselves. Therefore, instead of exploring new texture features or extensively studying existing features, we make use of well-known texture features (as studied, e.g., by J. S. Weszka *et al.* [17]), derived from the Fourier power spectrum, from co-occurrence statistics, and from gray level difference statistics.

5.1 Texture Features

5.1.1 Fourier power spectrum

The power spectrum $P(u, v)$ of image $I(i, j)$ is based on the Fourier transform of image $I(i, j)$

$$\mathcal{F}(u, v) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j) exp[-\frac{2\pi i}{N}(iu+jv)], u = 0, 1, ..., N-1, v = 0, 1, ..., N-1$$

such that

$$P(u,v) = \sqrt{(Re[\mathcal{F}(u,v)])^2 + (Im[\mathcal{F}(u,v)])^2}$$

Texture coarseness is indicated by the character of the distribution $P(u,v)$. A coarse texture will have high values concentrated near the origin $(u,v) = (0,0)$; conversely, a spread-out distribution $P(u,v)$ indicates a fine textural pattern. Similarly, a texture with major orientation θ will have high values of $P(u,v)$ concentrated around the perpendicular direction $\theta + \frac{\pi}{2}$. Thus the one-dimensional representation of texture coarseness may be taken as averages of $P(u,v)$ over annular rings or wedges centered at the origin.

The texture feature based on an annular ring with inner radius r_1 and outer radius r_2 is given by

$$P(r_1, r_2) = \sum_{r_1^2 \le u^2+v^2 \le r_2^2,\ 0 \le u,v \le N-1} |\mathcal{F}(u,v)|^2,$$

while the feature based on a wedge defined by angles θ_1 and θ_2 is of the form

$$P(\theta_1, \theta_2) = \sum_{\theta_1 \le \arctan(\frac{u}{v}) \le \theta_2,\ 0 < u,v \le N-1} |\mathcal{F}(u,v)|^2.$$

These two features are sensitive to size (the first) or orientation (the second) only, but not to both. In order to get features which are sensitive to both size and orientation, the power spectrum based on intersections of annular rings and of wedges was used.

5.1.2 Gray level difference statistics

Texture features based on gray level differences are derived from absolute differences between pairs of gray levels or average of gray levels. For any given displacement vector

$$\mathbf{d} = (\delta_x, \delta_y)$$

where both δ_x and δ_y are integers, the gray-level difference at displacement vector $\mathbf{d}$ is given by

$$f_{\mathbf{d}}(i,j) = |I(i,j) - I(i+\delta_x, j+\delta_y)|.$$

The probability density $p_{\mathbf{d}}$ of $f_{\mathbf{d}}(i,j)$ can be easily obtained by counting the number of occurrences of each value of $f_{\mathbf{d}}(i,j)$.

With a coarse texture such that the magnitude of $\mathbf{d}$ is relatively small with respect to the texture element size, the values of $p_{\mathbf{d}}$ tends to be concentrated near $i = 0$. Conversely, with a fine texture such that the magnitude of

$\mathbf{d}$ is comparable to the element size, the values of $p_{\mathbf{d}}$ tend to be spread out. The following four measures based on gray levels have often been suggested:

1. Contrast
$$CON = \sum i^2 p_{\mathbf{d}}(i)$$
(the moment of inertia around the origin).

2. Angular Second Moment
$$ASM = \sum p_{\mathbf{d}}^2(i)$$
Note that the ASM tends to be small when the $p_{\mathbf{d}}$ is relatively flat and larger otherwise.

3. Entropy
$$ENT = -\sum p_{\mathbf{d}}(i) \ln p_{\mathbf{d}}(i).$$
This value measures the uniformity of the density $p_{\mathbf{d}}$.

4. Mean
$$MEAN = E_{p_{\mathbf{d}}}(i)$$
This value is small when the $p_{\mathbf{d}}$ are concentrated near origin.

If a texture is directional, the degree of spread of values in above four measures should vary with the direction of $\mathbf{d}$.

5.1.3 Co-occurrence statistics

If there are n possible gray levels in image $I(i,j)$, then the $n \times n$ co-occurrence matrix $M_{\mathbf{d}}$ has entries $m_{i,j}$, where $m_{i,j}$ is determined by the observed frequency of occurrence of gray levels from i to j at separation $\mathbf{d}$. Similarly, the density function $p_{\mathbf{d}}(i,j)$ can be determined by the corresponding relative frequencies.

With a coarse texture such that the magnitude of $\mathbf{d}$ is small compared to the element size, the matrix $M_{\mathbf{d}}$ is nearly diagonal. With a fine texture, the matrix $M_{\mathbf{d}}$ has non-zero entries arbitrarily far from the diagonal. If a texture is directional, the degree of spread of values of $M_{\mathbf{d}}$ varies with direction $\mathbf{d}$.

We shall use three measures of coarseness of texture.

1. Contrast
$$CON = \sum (i-j)^2 p_{\mathbf{d}}(i,j)$$
This is the moment of inertia of the matrix around its main diagonal.

Table 1: Classification using the 9-NN method with unnormalized features

	1	2	3	4	5	6
1	20	6	6	0	0	0
2	6	19	2	5	0	0
3	10	3	19	0	0	0
4	1	10	0	20	0	1
5	0	1	0	4	22	5
6	0	0	0	2	8	22

2. Angular Second Moment

$$ASM = \sum p_{\mathbf{d}}^2(i,j)$$

The value will be small when $p_{\mathbf{d}}$ is relatively uniform.

3. Entropy

$$ENT = -\sum p_{\mathbf{d}}(i,j) \ln p_{\mathbf{d}}(i,j)$$

This value reflects the uniformity of $p_{\mathbf{d}}$.

5.2 Results

Our texture images (images of water, wool, fur, fiber, cork, and grass) are 512×512. To get both a training and a test set, we use 64×64 windows to partition each image, yielding 64 samples for each texture image. We randomly select 32 samples from each image to be training samples, reserving the other 32 samples as test samples. Furthermore, we use equal probability quantization to map the 256-level gray scale into a 64-level gray scale, in order to simplify our computational work.

From each sample we extract a total of 22 features. We use the power spectrum $P(r_1, r_2)$, for $(r_1, r_2) = (2,4), (4,8), (8,16), (16,31)$; nine features based on gray level differences, namely the CON, ASM and ENT measures with displacement vectors $d = (1,0), (0,1), (1,1)$; and nine features based on co-occurrence statistics, with the same choices as for gray levels. We used the k-nearest neighbor (k-NN) classification method as a measuring stick by which to evaluate our proposed classification method.

The k-NN classification method only performs well with normalized features, whereas our method does not exhibit such sensitivity. Using our

Table 2: Classification using the 9-NN method with normalized features

	1	2	3	4	5	6
1	28	1	2	0	1	0
2	1	27	2	1	0	1
3	3	2	26	0	1	0
4	2	4	0	25	0	1
5	1	0	0	1	27	3
6	0	0	0	1	2	29

Table 3: Classification using our proposed method

	1	2	3	4	5	6	7
1	25	2	3	1	0	1	0
2	4	24	1	2	0	1	1
3	2	2	26	0	0	0	2
4	0	3	2	25	0	0	2
5	0	1	0	3	24	4	0
6	0	0	0	0	3	29	0

classifier with threshold, we need to store 53 samples and obtain classification error rate of 19.8%. The k-NN classifier (with $k = 9$, as determined by experiments to produce the best possible results) yields a classification error rate of 36% with unnormalized features and of 15.6% with normalized features. In other words, our classifier easily outperforms the k-NN classifier with unnormalized features, but does slightly worse with normalized features. (In practice, normalization of features is a difficult task.)

The classification results with unnormalized features and normalized features are shown in Tables 6.1 through 6.3.

If we discard at each training stage that feature, the coefficient of which is less than 0.1 in the linear combination of normalized features, we obtain a subset of 11 features: the ENT and CON measures in gray level difference statistics, the CON measure in co-occurrence statistics, and $P(4, 8)$ and $P(16, 31)$. Restricting our experiments to these 11 features, we observe

Table 4: Classification using the 9-NN method with 11 normalized features

	1	2	3	4	5	6
1	30	0	1	0	1	0
2	1	29	1	1	0	1
3	3	3	26	0	0	0
4	0	7	0	25	0	0
5	1	0	0	0	28	3
6	0	0	0	0	2	30

Table 5: Classification using our proposed method with 11 features

	1	2	3	4	5	6	7
1	24	2	3	2	0	1	0
2	3	23	1	4	0	1	1
3	2	2	26	0	0	0	2
4	0	4	2	24	0	0	2
5	0	1	0	3	22	6	0
6	0	0	0	0	4	28	0

that the k-NN classifier (using normalized features) yields the reduced error rate of 12.5%, while our proposed method yields the somewhat increased error rate of 23%. (This somewhat curious outcome is presumably due to an indirect editing effect of the feature selection, which favors the nearest neighbor classifier—as it is otherwise easily misled—and mildly punishes our proposed classifier—by depriving it of a small amount of useful information.) Results are shown in Tables 6.4 and 6.5.

6 Summary and Conclusion

While this application is obviously limited in scope as well as somewhat compromised by an indifferent choice of features, it is nevertheless encouraging in that it demonstrates that our proposed method is competitive, both in terms of efficiency and of accuracy. In particular, the insensitivity of our classifier to feature scaling and relative insensitivity to dimensionality make it applicable in a broad range of practical applications. Moreover, our method also naturally leads to a good feature selection, a selection which can be used with other classifiers.

7 References

1. T.W. Anderson, "Some nonparametric multivariate procedures based on statistically equivalent blocks," in *Multivariate Analysis*, P. R. Krishnaiah, ed., pp. 5–27, Academic Press, New York., 1966

2. L. Breiman, W. Meisel and E. Purcell, "Variable kernel estimates of multivariate densities," *Technometrics* **19** (1977), pp. 135–144.

3. L. Devrove and L. Györfi. *Nonparametric Density Estimation: the L_1 view.* John Wiley & Sons, New York, 1985.

4. V.A. Epanechnikov, "Nonparametric estimates of a multivariate probability density," *Theoretical Probability Applications* **14** (1969), pp. 153–158.

5. J.H. Friedman, "A recursive partitioning decision rule for nonparametric classification," *IEEE Trans on Computers* **C-26** (1977), pp. 404–408.

6. E.G. Henrichon. Jr. and K.S. Fu, "A nonparametric partitioning procedure for pattern recognition," *IEEE Trans. on Computers* **C-18** (1969), pp. xxx–xxx.

7. Xiaobo Li and R.C. Dubes, "Tree classifier design with a permutation statistic," *Pattern Recognition* **19** (1986), pp. 229–235.

8. Y.P. Mack and M. Rosenblatt, "Multivariate k-nearest neighbor density estimate," *J. of Multivariate Analysis* **9** (1979), pp. 1–15.

9. H.J. Payne and W. Meisel, "An algorithm for constructing optimal binary decision trees," *IEEE Trans. on Computers* **C-26** (1977), pp. 905–916.

10. J.W. Raatgever and R.P.W. Duin, "On the variable kernel model for multivariate nonparametric density estimation," *Pattern Recognition Letters* **1** (1978), pp. 293–298.

11. E.M. Rounds, "A combined nonparametric approach to feature selection and binary decision tree," *Pattern Recognition* **12** (1980), pp. 313–317.

12. I.K. Sethi and G.P.R. Sarvarayudu, "Hierarchical classifier design using mutual information," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **PAMI-4** (1982), pp. 441–445.

13. J.L. Talmon, "A multiclass nonparametric partitioning algorithm," *Pattern Recognition Letters* **4** (1986), pp. 31–38.

14. C.T. Tseng. *The Design of a Nonparametric Classification System.*' Ph.D. dissertation, U. of New Mexico, 1989.

15. C.T. Tseng and B.M.E. Moret, "A method for the choice of smoothing parameters," accepted for publication in *Mathematical and Computer Modelling.*

16. C.T. Tseng and B.M.E. Moret, "A method for one-dimensional feature linear transformation," accepted for publication in *Pattern Recognition.*

17. J. Van Ness, "On the dominance of non-parametric Bayes rule discriminant algorithms in high dimensions," *Pattern Recognition* **12** (1980), pp. 355–368.

18. T.J. Wagner, "Nonparametric estimates of probability densities," *IEEE Transaction on information theory* **IT-21** (1975), pp. 438–440.

19. J.S. Weszka, C.R. Dyer and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Trans. On Systems, Man, and Cybernetics* **SMC-6** (1976), pp. xxx–xxx.