Classification based on hybridization of parametric and nonparametric classifiers

Probal Chaudhuri[1], Anil K. Ghosh[2] and Hannu Oja[3]

[1]**Theoretical Statistics and Mathematics Unit, Indian Statistical Institute, 203, Barrackpore Trunk Road, Calcutta 700108, India.**

[2]**Department of Mathematics and Statistics, Indian Institute of Technology, Kanpur 208016, India.**

[3] **Tampere School of Public Health, University of Tampere, Fin 33014, Finland.**

E-mail: probal@isical.ac.in, anilkghosh@rediffmail.com, hannu.oja@uta.fi

Abstract

Parametric methods of classification assume specific parametric models for competing population densities (e.g., Gaussian population densities lead to linear and quadratic discriminant analysis), and they work well when these model assumptions are valid. Violation in one or more of these parametric model assumptions often leads to a poor classifier. On the other hand, nonparametric classifiers (e.g., nearest neighbor and kernel based classifiers) are more flexible and free from parametric model assumptions. But statistical instability of these classifiers may lead to poor performance when we have small training samples. Nonparametric methods, however, do not use any parametric structure of population densities. So, even when one has some additional information about population densities, that important information is not used to modify the nonparametric classification rule. This article makes an attempt to develop some *hybrid classification* methods by combining the strength of parametric

and nonparametric approaches. We use some simulated examples and some benchmark data sets to examine the performance of these hybrid discriminant analysis tools. Asymptotic results on their misclassification rates have been derived under appropriate regularity conditions.

Index Terms : Bayes risk, kernel density estimation, kernel discriminant analysis, linear discriminant analysis, misclassification rate, multi-scale analysis, nearest neighbor density estimation, nearest neighbor classification, quadratic discriminant analysis.

1 Introduction

A popular approach in discriminant analysis is to estimate the population densities using the training data and then plug-in those estimates in the Bayes rule [33] to construct the classifier. If $\hat{f}_j$ is the estimated density function for the j-th population ($j = 1, 2, \ldots, J$) and π_j is its prior probability, the classification rule $\mathsf{d} : \Re^d \to \{1, 2, \ldots, J\}$ can be expressed as $\mathsf{d}(\mathbf{x}) = \text{argmax } \pi_j \hat{f}_j(\mathbf{x})$. When the π_js are not known, one usually estimates them using the proportion of class sizes in the training sample. The density estimate $\hat{f}_j$ $(j = 1, 2, \ldots, J)$ can be computed either parametrically or nonparametrically. In parametric approaches [1], [33], the f_js are assumed to be known except for a few parameters, and one uses the training data to estimate these parameters. For instance, in Fisher's [10] linear and quadratic discriminant analysis (LDA and QDA), the f_js are assumed to be normal with equal or unequal dispersion matrices, respectively. Clearly, the performance of these parametric classifiers depends on the validity of these model assumptions. If the f_js are close to the assumed parametric models, they are expected to perform very well. But violation in one or more assumptions

may lead to poor classification by these parametric procedures.

Nonparametric classifiers [9], [21], on the other hand, are more flexible and free from such parametric model assumptions. Kernel discriminant analysis [19], [40] and nearest neighbor classification [11], [7] are two well known examples of nonparametric classifiers that use the kernel method [40], [37], [43] and the nearest neighbor method [30], [14] for density estimation, respectively. One can also use other nonparametric density estimates [37], [8], [27] for classification. However, in classification, one is more interested in estimates of class boundaries rather than estimates of population density functions. Classification trees [2], neural nets [35] and support vector machines [42] are some of the nonparametric classifiers that directly estimate the class boundaries without going for density estimation. Note that, none of these nonparametric methods uses any parametric structure of the population densities. However, when one has some insight or additional information about the population distributions that can be modeled parametrically, that information should be used to modify nonparametric classification rules. This is important because when the population distributions happen to be close to the assumed parametric models, nonparametric methods are expected to have significantly higher misclassification rates than those of parametric classifiers.

Both parametric and nonparametric methods have their own strength and limitations. This paper aims to combine the strength of these two methods to develop some classification techniques. Here, we assume a parametric model $f_j(\mathbf{x}) = f_j^0(\mathbf{x}, \boldsymbol{\beta}_j)$ $(= f_j^0(\mathbf{x})$, say) for the j-th class $(j = 1, 2, \ldots, J)$ to start with and estimate the unknown parameter $\boldsymbol{\beta}_j$ (which may be a real or a finite dimensional vector valued parameter) to get the initial parametric

density estimate $\hat{f}_j^0(\mathbf{x}) = f_j^0(\mathbf{x}, \hat{\boldsymbol{\beta}}_j)$. Note that if the true population density f_j is far from the assumed parametric model, the use of $\hat{f}_j^0$ is likely to lead to poor classification. So, one needs to make some adjustment to these parametric density estimates to guard against it. Here, we use a nonparametric adjustment factor ($\theta_j : \Re^d \to \Re_+$) for this purpose, and the final density estimate is obtained as $\hat{f}_j^*(\mathbf{x}) = \theta_j(\mathbf{x})\hat{f}_j^0(\mathbf{x})$ (see also [22], [26], [23] for discussion on nonparametric adjustment to parametric density estimates). When the f_j^0s ($j = 1, 2, \ldots, J$) are close to the true densities f_j, the θ_js are expected to be very close to 1 over the entire measurement space (different formulae for θ_js are proposed in Section 2), and in that case, the hybrid method behaves like the associated parametric classifier. As a consequence, the resulting classifier here is expected to perform better than fully nonparametric methods. Again, when the true densities are far from the assumed parametric models, the nonparametric adjustment factors provide a safeguard against deviations from parametric model assumptions by making significant adjustment to the initial parametric density estimates. As a result, the performance of the classifier gets much improved and comparable to that of the nonparametric classifiers in such cases.

Like other nonparametric methods, the adjustment factors θ_js ($j = 1, 2, \ldots, J$), and hence the misclassification rate, depend on the values of the associated smoothing parameters. One should note that in addition to depending on the entire training sample, a good choice of smoothing parameters depends on the specific observation to be classified. A fixed level of smoothing may not work well in all parts of the measurement space. Therefore, in practice, it would be useful to study the classification results for multiple levels of smoothing simultaneously. This type of multi-scale analysis was carried out in [4], [5], [18] in the con-

text of function estimation. Some other authors [15], [17], [16] adopted similar multi-scale techniques for visualization of classification results over a wide range of smoothing parameters. Results obtained at different scales of smoothing can be combined judiciously to arrive at the final decision. This aggregation method is similar in spirit with classifiers based on bagging [3], boosting [36] and other popular ensemble methods.

2 Description of hybrid classifiers

As it has been mentioned before, here we start with a parametric model that assumes $f_j(\mathbf{x}) = f_j^0(\mathbf{x}, \boldsymbol{\beta}_j)$ $(j = 1, 2, \ldots, J)$ and estimate the parameter $\boldsymbol{\beta}_j$ using the training data. This leads to the initial parametric density estimate $\hat{f}_j^0(\mathbf{x}) = f_j^0(\mathbf{x}, \hat{\boldsymbol{\beta}}_j)$. Then, a nonparametric adjustment factor θ_j is used to modify $\hat{f}_j^0(\mathbf{x})$ and to get the final density estimate $\hat{f}_j^*(\mathbf{x})$. In that sense, our density estimates are hybrids of parametric and nonparametric methods, and that is why we call the resulting classifier a hybrid classifier. Hjort and Glad [22] proposed one such hybrid density estimate using an adjustment factor based on the kernel method. A similar hybrid density estimate was considered in [26], where instead of starting with a parametric model, the authors used general density estimates. Hjort and Jones [23] proposed another nonparametric adjustment based on local likelihood criterion. These types of density estimation methods are also discussed in [34] and [24]. Though all these density estimates are based on the kernel method, one can borrow these ideas to develop their nearest neighbor versions as well. In this article, we use both kernel and nearest neighbor methods to construct some hybrid classifiers, which are given below.

2.1 Hybridization of kernel and parametric density estimates

Here, we propose three different types of adjustment factors for the construction of $\hat{f}_j^*(\mathbf{x})$ and the classification rule $\mathsf{d}^*(\mathbf{x}) = \text{argmax}\ \pi_j \hat{f}_j^*(\mathbf{x})$.

Method-1 : Along with the parametric density estimate $\hat{f}_j^0(\mathbf{x})$ $(j = 1, 2, \ldots, J)$, we consider the nonparametric kernel density estimate [40], [37], [43] $\hat{f}_{jh_j}^1(\mathbf{x})$, which is given by

$$\hat{f}_{jh_j}^1(\mathbf{x}) = n_j^{-1} h_j^{-d} \sum_{i=1}^{n_j} K\{h_j^{-1}(\mathbf{x} - \mathbf{x}_{ji})\},$$

where $\mathbf{x}_{j1}, \mathbf{x}_{j2}, \ldots, \mathbf{x}_{jn_j}$ are observations from the j-th population, K is a d-dimension density function symmetric about $\mathbf{0}$ and $h_j > 0$ is the associated smoothing parameter commonly known as the bandwidth. Several choices for the kernel function K are available in the literature [40]. Throughout this article, we shall use the Gaussian kernel $K(\mathbf{t}) = (2\pi)^{-d/2} e^{-\mathbf{t}'\mathbf{t}/2}$ for our purpose. To compute the adjustment factor $\theta_j(\mathbf{x})$, we consider a closed ball $B(\mathbf{x}, r_j) = \{\mathbf{y} : \rho(\mathbf{x}, \mathbf{y}) \leq r_j\}$ of radius r_j around $\mathbf{x}$ and calculate the probability measures of that ball $P\{\mathbf{X} \in B(\mathbf{x}, r_j)\}$ under both the estimated parametric and the nonparametric models. If we denote these two probabilities by $P^0(\mathbf{x}, r_j)$ and $P^1(\mathbf{x}, r_j)$, respectively, the adjustment factor θ_j is given by $\theta_j(\mathbf{x}, r_j) = P^1(\mathbf{x}, r_j)/P^0(\mathbf{x}, r_j)$. So, the final density estimate at $\mathbf{x}$ is obtained as $\hat{f}_{jh_j}^{(1)}(\mathbf{x}) = \hat{f}_j^0(\mathbf{x}) P^1(\mathbf{x}, r_j)/P^0(\mathbf{x}, r_j)$. The radius of the ball r_j behaves like a smoothing parameter here. Greater the value of r_j, θ_j tends to be smoother in some sense. Note that, if the r_js $(j = 1, 2, \ldots, J)$ are small, for all $j = 1, 2, \ldots, J$, $P^1(\mathbf{x}, r_j)/P^0(\mathbf{x}, r_j)$ will be close to $\hat{f}_{jh_j}^1(\mathbf{x})/\hat{f}_j^0(\mathbf{x})$, and in that case, the resulting classifier will behave like the associated nonparametric classifier. On the other hand, for large values of r_j, both P^1 and P^0 will be close to 1, and so will be the ratio P^1/P^0. Therefore, in that case, the hybrid classifier will behave like the associated parametric classifier.

In addition to $r_j s$ $(j = 1, 2, \ldots, J)$, there is another set of smoothing parameters h_js involved in this method. Instead of dealing with these two sets of smoothing parameters simultaneously, for computational simplicity, we take $r_j = 3h_j$ for all $j = 1, 2, \ldots, J$. Note that if K is Gaussian, the observations $\mathbf{y}$ with $\|\mathbf{y} - \mathbf{x}\|/h_j > 3$ have negligible effect on $\hat{f}^1_{jh_j}(\mathbf{x})$. This motivated us to take $r_j = 3h_j$. The use of the same bandwidth h_j in all directions requires some preliminary transformation of the data (i.e., sphering of the data). Here, we have used the usual moment based estimate of the dispersion matrix to make that transformation.

Method-2 : A hybrid density estimate that is constructed by using nonparametric adjustment of a parametric density estimate was proposed in [22]. This density estimate can be expressed as

$$\hat{f}^{(2)}_{jh_j}(\mathbf{x}) = \hat{f}^0_j(\mathbf{x}) \times \frac{1}{n_j}\sum_{i=1}^{n_j} h_j^{-d} K\{h_j^{-1}(\mathbf{x} - \mathbf{x}_{ji})\}\frac{1}{\hat{f}^0_j(\mathbf{x}_{ji})} = \frac{1}{n_j}\sum_{i=1}^{n_j} h_j^{-d} K\{h_j^{-1}(\mathbf{x} - \mathbf{x}_{ji})\}\frac{\hat{f}^0_j(\mathbf{x})}{\hat{f}^0_j(\mathbf{x}_{ji})}.$$

As we have mentioned before, a similar kind of nonparametric adjustment factor was also proposed in [26] but instead of taking a parametric model for initial density estimation, the authors considered a general set up. One should note that the term $\hat{f}^0_j(\mathbf{x})/\hat{f}^0_j(\mathbf{x}_{ji})$ can be very influential when $\hat{f}^0_j(\mathbf{x}_{ji})$ is close to zero but $\hat{f}^0_j(\mathbf{x})$ is away from it. To get rid of such situations, following [22], we have truncated the values of these ratios so that $0.1 \leq \hat{f}^0_j(\mathbf{x})/\hat{f}^0_j(\mathbf{x}_{ji}) \leq 10$.

Method-3 : The adjustment factor $\theta_j(\mathbf{x})$ can also be obtained by maximizing the local loglikelihood score [23] $L\{\theta_j(\mathbf{x})\} = \hat{f}^1_{jh_j}(\mathbf{x})$ log $\theta_j(\mathbf{x})$ - $\theta_j(\mathbf{x})$ $h_j^{-d} \int K\{h_j^{-1}(\mathbf{x} - \mathbf{t})\}\hat{f}^0_j(\mathbf{t})d\mathbf{t}$. The resulting density estimates are given by

$$\hat{f}^{(3)}_{jh_j}(\mathbf{x}) = \hat{f}^1_{jh_j}(\mathbf{x})\hat{f}^0_j(\mathbf{x})/(K_{h_j} * \hat{f}^0_j)(\mathbf{x}), \qquad j = 1, 2, \ldots, J,$$

where $K_h(\mathbf{t}) = h^{-d}K(\mathbf{t}/h)$, and '$*$' denotes the convolution. If the assumed parametric model f_j^0 and the kernel K both are Gaussian, so is the convolution, and in that case, $K_{h_j} * \hat{f}_j^0$ has a nice closed form expression.

2.2 Hybridization of nearest neighbor and parametric density estimates

We adopt similar ideas to find the adjustment factors and hence the hybrid density estimates based on the nearest neighbor method.

Method-1 : Just like the kernel method, here also we use the adjustment factor P^1/P^0 but the neighborhoods are chosen in a different way. Instead of considering different balls (neighborhoods) for different classes, like usual nearest neighbor classifiers [11], [6], here we use the same ball for all populations. The distance between $\mathbf{x}$ and its k-th nearest neighbor in the training sample is taken as the radius of that ball. Let us denote this k-th nearest neighbor of $\mathbf{x}$ by $\mathbf{x}^{(k,n)}$ and the ball by $B_{n,k}(\mathbf{x}) = \{\mathbf{y} : \rho(\mathbf{x}, \mathbf{y}) \leq \rho(\mathbf{x}, \mathbf{x}^{(k,n)})\}$. Then, the resulting hybrid density estimates are given by

$$\hat{f}_{j,k}^{(1)}(\mathbf{x}) = \hat{f}_j^0(\mathbf{x}) \frac{k_j/n_j}{P_{f_j}\{\mathbf{X} \in B_{n,k}(\mathbf{x})\}} \qquad j = 1, 2, \ldots, J,$$

where k_j ($\sum k_j = k$) is the number of observations from class-j in $B_{n,k}(\mathbf{x})$, and n_j ($\sum n_j = n$) is the training sample size for the j-th class. These density estimates depend on the distance function ρ, and throughout this article, we use the Euclidean distance for this purpose. If the measurement variables are not of comparable units and scales, using Euclidean metric is not a sensible option. Therefore, in our data analysis, we standardized the observations

using the usual moment based estimate of the pooled dispersion matrix before using the Euclidean metric. This is equivalent to using the Mahalanobis distance [32]. However, one may use other flexible or adaptive metrics [12], [20] as well. Unlike the kernel based hybrid methods, here the hybrid density estimates and hence the resulting classifier depend on a single smoothing parameter k. One may use different balls for different populations, and for the j-th class, one can take the radius of the ball as a function of k_j to make the classifier dependent on several smoothing parameters. However, we do not consider those in this paper.

Method-2 : Though no hybrid density estimate based on nearest neighbor method was proposed in [22], such an extension is possible. Once again, we use the same neighborhood $B_{n,k}(\mathbf{x})$ for all populations, and the resulting density estimates can be expressed as

$$\hat{f}^{(2)}_{j,k}(\mathbf{x}) = \frac{\hat{f}^0_j(\mathbf{x})}{n_j Vol\{B_{n,k}(\mathbf{x})\}} \sum_{x_{ji} \in B_{n,k}(\mathbf{x})} 1/\hat{f}^0_j(\mathbf{x}_{ji}) \qquad\qquad j = 1, 2, \ldots, J.$$

To avoid high sensitivity of $\hat{f}^0_j(\mathbf{x})/\hat{f}^0_j(\mathbf{x}_{ji})$ near zero value of the denominator, we have truncated these ratios so that they take values in the interval $[0.1, 10]$ as before.

Method-3 : Straightforward conversion for the local loglikelihood method into its nearest neighbor version is not feasible because of the absence of any meaningful analog for the convolution part in the denominator of $\hat{f}^{(3)}_{jh_j}(\mathbf{x})$. However, one may look at the convolution as the expectation of the kernel density estimate under $\hat{f}^0_j$. So, it can be replaced by the expectation of the nearest neighbor density estimate to get an analogous version.

$$\hat{f}^{(3)}_{j,k}(\mathbf{x}) = \hat{f}^0_j(\mathbf{x}) \frac{k_j/n_j V_{n,k}(\mathbf{x})}{E_{\hat{f}^0}\{k_j/n_j V_{n,k}(\mathbf{x})\}} = \hat{f}^0_j(\mathbf{x}) \frac{k_j/V_{n,k}(\mathbf{x})}{E_{\hat{f}^0}\{k_j/V_{n,k}(\mathbf{x})\}},$$

where $V_{n,k}(\mathbf{x}) = Vol\{B_{n,k}(\mathbf{x})\}$, and $E_{\hat{f}^0}$ denotes expectation over the whole training sample, where $\mathbf{x}_{ji} \sim \hat{f}_j^0$ for all $j = 1, 2, \ldots, J$ and $i = 1, 2, \ldots, n_j$. Unlike the kernel method, here the denominator of $\hat{f}_{j,k}^{(3)}$ may not have a closed from expression. One can approximate this by an empirical average computed using repeated generations of observations from the initial parametric distributions.

2.3 Multi-scale approach in formation of final classifier by weighted averaging

The adjustment factor $\theta_j(\mathbf{x})$ $(j = 1, 2, \ldots, J)$ depends on the associated smoothing parameter for each of the methods described in Sec. 2.1 and 2.2. In classification problems, a good choice of the smoothing parameter not only depends on that class but also on other competing class densities. Therefore, instead of looking at the accuracy of the hybrid density estimates separately for different classes, in a classification problem, it is more meaningful to consider all density estimates corresponding to different classes simultaneously. But, when there are several competing classes, finding a good set of smoothing parameters is computationally infeasible as there would be different smoothing parameters associated with different density estimates corresponding to different classes. To reduce this computational cost, in the case of kernel based methods, we have used the same bandwidth h for all populations after standardization of observations using the usual moment based estimate of pooled dispersion matrix. Note that in the case of nearest neighbor based methods, our construction of hybrid classifiers (as discussed in Section 2.2) allows the method to depend only on one smoothing parameter k. There also, we have standardized the observations in the same way.

One can use cross-validation [29], [41] or some other method to find out the optimum value of the smoothing parameter like bandwidth of a kernel or the number of nearest neighbors. Then, that can be used for classification of all observations. However, one should note that in addition to depending on the training sample, a good choice of the smoothing parameter depends on the specific observation to be classified. A fixed level of smoothing may not work well in all parts of the measurement space. Therefore, instead of fixing the value of the smoothing parameter, it may be of more use to simultaneously study the classification results for all scales of smoothing in some appropriate range. The usefulness of this multi-scale smoothing has been discussed in the literature by many authors both in the context of function estimation [4], [5], [18] and classification [15], [17]. Results obtained for different levels of smoothing can be aggregated in a judicious way to arrive at the final classifier. A natural way to aggregate these results is to take weighted average of posterior probabilities. So, the aggregated final classifier can be expressed as

$$\mathsf{d}_A(\mathbf{x}) = \arg\max_j \sum_s W(s) \left\{ \frac{\pi_j \hat{f}^*_{js}(\mathbf{x})}{\sum_{t=1}^{J} \pi_t \hat{f}^*_{ts}(\mathbf{x})} \right\},$$

where s is the smoothing parameter, $W(s)$ is the weight function ($\sum W(s) = 1$), and $\hat{f}^*_{js}$ is the hybrid density estimate for the j-th class ($j = 1, 2, \ldots, J$). Note that the bandwidth of a kernel h is a continuous smoothing parameter, and it is not possible to use all values of h lying in an interval. Instead, for data analysis, we use some discrete values in that range.

2.4 Large sample properties

From consistency results on kernel density estimates [40], [37], [43] and nearest neighbor density estimates [30], we know that under certain regularity conditions (different sets of

conditions for kernel and nearest neighbor methods are required), both of these two nonparametric density estimates converge (in probability) to the true density function. Consistency of these hybrid density estimates under similar regularity conditions (see Propositions 1 and 2 in the Appendix) follows from these results. The asymptotic optimality of the error rates of our hybrid classifiers follows from the consistency of these hybrid density estimates. This result is formally presented in the following theorem, and the proof is given to the Appendix.

Theorem 1*: Suppose that for all* $j = 1, 2, \ldots, J$, $f_j^0(\mathbf{x}, \boldsymbol{\beta}_j)$ *is continuous in* $\mathbf{x}$ *and* $\boldsymbol{\beta}_j$. *Define* $\hat{\boldsymbol{\beta}}_j$ *and* $\boldsymbol{\beta}_j^0$ *as in Propositions 1 and 2 given in the Appendix and assume that* $\hat{\boldsymbol{\beta}}_j$ *is a consistent estimate for* $\boldsymbol{\beta}_j^0$ *for all* $j = 1, 2, \ldots, J$. *Also, define* L_n *and* U_n *as the lower and the upper bounds for the smoothing parameter when* n *is the training sample size.*

(a) Assume that L_n *and* U_n *both converge to* 0 *and* nL_n^d *and* nU_n^d *both tend to* ∞ *as* $n \to \infty$. *Then, the misclassification rate of the kernel based aggregated hybrid classifier converges to the optimal Bayes risk for both of Method-2 and Method-3. If the kernel function* K *has bounded variation and the* f_j*s are uniformly continuous, the above result holds also for Method-1 if* $nL_n^d/log(n)$ *and* $nU_n^d/log(n)$ *both tend to* ∞ *as* $n \to \infty$

(b) Suppose that L_n *and* U_n *both tend to* ∞ *and* L_n/n *and* U_n/n *both converge to* 0 *as* $n \to \infty$. *Then, misclassification rates of all of Method-1, Method-2 and Method-3, which are aggregated hybrid classifiers based on nearest neighbor techniques, converge to the optimal Bayes risk.*

2.5 Choice of the weight function

In practice it is necessary to find the upper and the lower limits (U_n and L_n as in Theorem 1) of the smoothing parameters for aggregation. For the kernel based methods, we have followed the idea of [17] to set these limits. After standardizing the observations, we compute all pairwise distances between the standardized observations of a class and then calculate the lower and the upper 5-th percentiles ($\lambda_{0.05}(j)$ and $\lambda_{0.95}(j),\ j = 1, 2, \ldots, J$) from it. For our multi-scale analysis, we have taken a conservative approach and set $\min\{\lambda_{0.05}(j)/3\}$ and $\max\{\lambda_{0.95}(j)\}$ as the lower and the upper limit, respectively. The choice of the factor '1/3' is motivated by the use of Gaussian kernel function (see [17] for a detail discussion on the choice of upper and lower limits). For nearest neighbor based methods, we have taken a simplified approach of considering $k = 1, 2, .., n-1$.

Note that the asymptotic optimality of the hybrid classifiers does not depend on the weight function W. It is transparent from Theorem 1 that as long as $W(\cdot) > 0$ only in the range $[L_n, U_n]$ and $\sum W(\cdot) = 1$, error rates of the hybrid classifiers converge to the Bayes risk. However, in practice, a suitable weight function should be chosen for aggregation. One would naturally rely more on the classifiers which lead to lower misclassification rates. So, the weight function should be a decreasing function of the misclassification rate $\Delta(s)$. Popular aggregation methods (e.g., boosting, where $W = log\{(1-\Delta)/\Delta\}$ is taken as the weight function [36], [13]) use similar ideas for aggregation. However, it is our empirical experience that the log function used in boosting decreases with Δ at a very slow rate, and it fails to appropriately weigh down the poor classifiers resulting from poor choices of the smoothing parameter. So, for our data analysis, we have used the weight function proposed

in [15] and [17], which is given by

$$W(s) = Ce^{-\frac{1}{2}\left\{\frac{\Delta(s)-\Delta_\circ}{\sqrt{\Delta_\circ(1-\Delta_\circ)/n}}\right\}^2},$$

where $\Delta_\circ = \min_s \Delta(s)$, C is a normalizing constant, and all misclassification rates are estimated by leave-one-out cross-validation technique [29]. This Gaussian-type weight function decreases at a faster rate, and it is expected to weigh down the poor classifiers appropriately. This choice of the weight function worked reasonably well in our examples, which we will see later in Sections 3 and 4. Our empirical experience also suggests that the final result is not very sensitive on the choice of the weight function if it decreases appropriately at an exponential or higher order polynomial rate. One can get an intuitive feeling about it from Theorem 1, which guarantees the asymptotic optimality of the error rates of the hybrid classifiers for a wide range of weight functions.

3 Results on simulated examples

In this section, we use some simulated data sets to illustrate the performance of the proposed hybrid methods. For each of these data sets, we run different parametric, nonparametric and hybrid classifiers and report their error rates. For the sake of simplicity, here we restrict ourselves to two-class problems involving bivariate data. In the next section, we will present some classification results for high dimensional benchmark data sets involving more than two classes.

For each of these simulated examples, we take equal number of observations from the two classes to generate 250 different training and test sets, each of size 100 and 200, respectively.

Average test set error rates for the hybrid classifiers (both based on kernel and nearest neighbors) over those 250 trials are reported in Tables 1 and 2 alongwith their corresponding standard errors. For hybrid method-1, it is often difficult to get a closed form expression for P^1 and P^0. In such cases, one can generate observations from appropriate distributions to approximate these probabilities by Monte Carlo method. Here we used 5000 Monte Carlo replications for this purpose. However, like the adjustment factor in method-2, this approximated probability ratio can be very influential when P^0 is close to zero. It may happen when the observation is an outlier w.r.t. the assumed parametric model or the neighborhood size is very small. To cope with such situations, we used this ratio for density adjustment only when the approximated P^0 is grater than 0.001, otherwise the nonparametric density estimate f_j^1 was used as the final density estimate f_j^*. Error rates are also reported for LDA, QDA and nonparametric (kernels and nearest neighbors) classifiers. Note that both of the kernel discriminant analysis (KDA) and the nearest neighbor (NN) classifier require the optimum value of the smoothing parameter to be estimated. Here, we used leave-one-out cross validation method [41] for this purpose. Leave-one-out method sometime leads to multiple minimizers for the estimated error rate due to its stepwise nature. Here, we considered the smallest and the largest among this set of optimizers and reported the result for that one, which had lower test set error rate. To facilitate the comparison, Bayes error rates are reported as well. Throughout this section, prior probabilities of the two classes are taken to be equal.

We begin with an example involving Gaussian distributions, where the two populations $N_2(0, 0, 0.25, 0.25, 0)$ and $N_2(1, 1, 1, 1, 0)$ differ both in location and scatter parameters (let

us call it Example-1). We know that it is an ideal set up for QDA. So, as expected, it had the best performance in this case, and its average error rate was very close to the optimum Bayes risk (see Table 1). LDA led to the highest error rate in this example. Nonparametric classifiers, KDA and NN also had significantly higher misclassification rates than that of QDA. But when we started with the right parametric model (i.e., Gaussian distributions with unequal dispersion matrices), hybrid classifiers could achieve much better performance. All hybrid methods led to significantly lower misclassification rates than nonparametric classifiers. Among the hybrid methods, method-1 and method-3 performed better than method-2, and error rates of these two classifiers were not significantly different from that of QDA.

Next we consider two cases where the population distributions are asymmetric in nature. In Example-2, the measurement variables in each class are independent and identically distributed as lognormal variate, whereas in Example-3, lognormal distributions are replaced by exponential distributions. The parameters of the two populations in these two cases are given below

Example-2. $f_1 : X_1, X_2$ are i.i.d. $lognormal(0, 1)$ $f_2 : X_1, X_2$ are i.i.d. $lognormal(2, 1)$.

Example-3. $f_1 : X_1, X_2$ are i.i.d. $exp(0, 1)$ $f_2 : X_1, X_2$ are i.i.d. $exp(1, 1)$.

Since the population distributions were very different from normal, in these two examples, LDA and QDA did not perform well (see Table 1). Both these methods had more than 23% error rates in Example-3. In Example-2, QDA had an error rate of 13.25%, but for LDA, it was 18.76%. As compared to LDA and QDA, the nonparametric methods (KDA and NN) had significantly better performance. However, when we chose the right parametric model to start with, all hybrid methods significantly outperformed LDA, QDA and their nonparametric

counterparts. In these examples, overall performance of method-3 was somewhat better than the other two hybrid classifiers for both the kernel and the nearest neighbor hybridizations.

Table 1 : Average test set misclassification rates (in percentage) of different classifiers on simulated data sets and their standard corresponding errors

	Example-1	Example-2	Example-3
Bayes risk	13.31	7.87	6.77
LDA	16.22 (0.17)	18.76 (0.22)	23.09 (0.21)
QDA	13.63 (0.16)	13.25 (0.19)	23.01 (0.29)
KDA	15.34 (0.18)	11.43 (0.16)	12.66 (0.18)
NN	15.74 (0.20)	10.40 (0.16)	13.38 (0.18)
Hybridization based on kernels			
Method-1	13.65 (0.16)	10.64 (0.15)	10.39 (0.17)
Method-2	14.99 (0.16)	9.07 (0.14)	8.98 (0.15)
Method-3	13.69 (0.16)	8.12 (0.13)	8.42 (0.14)
Hybridization based on nearest neighbors			
Method-1	13.67 (0.17)	8.26 (0.12)	8.72 (0.15)
Method-2	14.47 (0.17)	8.50 (0.13)	9.05 (0.15)
Method-3	13.65 (0.17)	8.21 (0.12)	8.41 (0.14)

From Table 1, it is quite transparent that if we start with the right parametric models, hybrid method can achieve significantly better performance than their nonparametric counter parts. So, when one has some insight about the population densities that can be modeled parametrically, it is always advantageous to use hybrid classifiers. However, one may argue that it is better to use parametric classifiers in such cases. But one should note that parametric methods are sensitive to model mis-specification. In practice, validity of model assumptions is difficult to verify, and improper parametric models may lead to poor

classification by the parametric classifiers. But the nonparametric adjustment factor used in hybrid classification provides an automatic safeguard against it. So, even when the true population density functions are far from the assumed parametric models, unlike parametric methods, hybrid classifiers are capable to perform well and to achieve the performance of the nonparametric classifiers.

To illustrate the above point, once again we consider Examples 2 and 3, where the distributions are far from normal, and it is somewhat difficult to fit parametric models to these distributions. To study the robustness of the hybrid methods against parametric model mis-specification, here instead of starting with the true parametric models, we use Gaussian density functions with equal or unequal scatter matrices as the initial parametric models. Recall that LDA had poor performance in both of these examples, and QDA also had significantly higher error rate in Example-3. But, in spite of starting with wrong parametric models, hybrid methods did a reasonably good job. Most of the hybrid classifiers significantly improved the performance of LDA and QDA, and their error rates were comparable to those of the corresponding nonparametric methods. Only the kernel version of method-1 had comparatively higher error rates in Example-2, when we used the same scatter matrix for initial parametric models of different populations, but even in that case, its performance was significantly better than that of the corresponding parametric classifier. Clearly, unlike the parametric methods, the performance of the hybrid classifiers was not much affected by the wrong choice of the initial parametric models. To make it more transparent, we choose two other examples, where each of the two populations is an equal mixture of two Gaussian distributions.

Table 2 : Average test set misclassification rates (in percentage) of different classifiers simulated data sets and their corresponding standard errors

	Example-2	Example-3	Example-4	Example-5
Bayes risk	7.87	6.77	1.11	5.10
LDA	18.76 (0.22)	23.09 (0.21)	50.03 (0.13)	44.17 (0.16)
QDA	13.25 (0.19)	23.01 (0.29)	5.24 (0.10)	43.57 (0.15)
KDA	11.43 (0.16)	12.66 (0.18)	5.35 (0.10)	16.83 (0.22)
NN	10.40 (0.16)	13.38 (0.18)	5.50 (0.10)	17.94 (0.21)
Hybridization based on kernels : $f_j^0 = N(\mu_j, \mathbf{\Sigma})$				
Method-1	16.69 (0.23)	13.45 (0.24)	5.19 (0.10)	16.62 (0.22)
Method-2	10.25 (0.14)	12.40 (0.20)	5.19 (0.10)	16.85 (0.20)
Method-3	12.21 (0.20)	12.44 (0.21)	5.30 (0.10)	16.75 (0.21)
Hybridization based on kernels : $f_j^0 = N(\mu_j, \mathbf{\Sigma}_j)$				
Method-1	13.05 (0.20)	13.93 (0.24)	5.20 (0.10)	16.59 (0.22)
Method-2	10.89 (0.15)	12.33 (0.20)	5.23 (0.10)	16.94 (0.19)
Method-3	12.47 (0.18)	12.79 (0.22)	5.19 (0.10)	16.86 (0.21)
Hybridization based on nearest neighbors : $f_j^0 = N(\mu_j, \mathbf{\Sigma})$				
Method-1	10.35 (0.14)	13.53 (0.18)	5.16 (0.09)	17.10 (0.21)
Method-2	10.46 (0.14)	12.58 (0.17)	5.10 (0.10)	17.42 (0.21)
Method-3	10.52 (0.15)	13.47 (0.18)	5.13 (0.10)	17.00 (0.21)
Hybridization based on nearest neighbors : $f_j^0 = N(\mu_j, \mathbf{\Sigma}_j)$				
Method-1	10.92 (0.15)	13.93 (0.18)	5.17 (0.10)	17.30 (0.21)
Method-2	10.75 (0.15)	12.62 (0.18)	5.19 (0.10)	17.46 (0.21)
Method-3	11.26 (0.15)	13.89 (0.18)	5.18 (0.10)	17.25 (0.22)

Example-4. $f_1 : \{N_2(1, 1, 0.25, 0.25, 0) + N_2(-1, -1, 0.25, 0.25, 0)\}/2,$

$f_2 : \{N_2(1, -1, 0.25, 0.25, 0) + N_2(-1, 1, 0.25, 0.25, 0)\}/2.$

Example-5. $f_1 : \{N_2(1, 1, 0.25, 0.25, 0) + N_2(3, 3, 0.25, 0.25, 0)\}/2,$

$f_2 : \{N_2(2, 2, 0.25, 0.25, 0) + N_2(4, 4, 0.25, 0.25, 0)\}/2.$

Once again, LDA had very poor performance in both these examples. Also, the performance of QDA was not satisfactory in Example-5. But, again, inspite of starting with wrong parametric models, hybrid methods could lead to substantial improvement in misclassification rates. If not better, their performance was comparable to those of the nonparametric classifiers.

4 Results from the analysis of benchmark data sets

In this section, we analyze some benchmark data sets for further illustration of the proposed methods. Two of these data sets, namely the synthetic data and the satellite image (satimage) data, have specific training and test sets. For these two data sets, we report the test set error rates of different classifiers (see Table 4). When a classifier led to an error rate Δ, its standard error was computed as $\sqrt{\Delta(1-\Delta)/N_t}$, where N_t is the size of the test set. For the other data sets, which do not have specific training and test sets, we formed these sets by randomly partitioning the data into two parts. Sizes of the training and the test sets in each partition are reported in Table 3. This random partitioning was carried out 250 times to generate 250 different training and test samples. Average test set misclassification rates of different methods and their corresponding standard errors were computed over these 250

trials, and they are reported in Table 4. Among the data sets that we used in this section, salmon data was taken from [25]. The rest of the data sets and their descriptions are available either at UCI Machine Learning Repository (http://www.ics.uci.edu/∼mlearn) or at CMU Data Archive (http://lib.stat.cmu.edu). A brief description of these data sets is given in Table 3 below. Throughout this section, we used Gausssian distributions as the initial parametric models, and proportions of different classes in the training sample were used as their prior probabilities.

Table 3 : Brief description of benchmark data sets

Datasets	Salmon	Biomed	Iris	Diabetes	Crab	Wine	Synthetic	Satimage
No. of Classes	2	2	3	3	4	3	2	6
No. of Variables	2	4	4	5	5	13	2	4*
Training sample size	50	100	75	75	100	90	250	4435
Test sample size	50	94	75	70	100	88	1000	2000

* Only for central pixel values were used for classification.

We know that in the case of salmon data, population distributions are nearly Gaussian. So, both LDA and QDA performed well, and their performance was much better than nonparametric classifiers (see Table 4). Hybrid classifiers also took the advantage of starting with Gaussian models, which are nearly the correct models in this case. All hybrid methods had error rates similar to parametric methods and significantly better than their nonparametric counter parts. In the case of biomedical data, once again parametric methods, especially QDA, performed significantly better than nonparametric classifiers. This gives an indication that Gaussian distribution with different scatter matrices may fit well for this data. So, it was no surprise to see much better performance by the hybrid classifiers when we started

with these parametric models. Even the use of the same scatter matrix led to significantly lower error rates as compared to those of the nonparametric classifiers.

Like the salmon data, in the case of Iris data, population distributions are nearly Gaussian, but they are more separated. So, here all classifiers had almost similar error rates. This is a dataset where LDA is known to perform well, and it is difficult for other methods to beat LDA. Most of the hybrid methods had error rates similar to that of LDA, especially when we started with Gaussian distributions with the same scatter matrix for different populations. On the diabetes data, the performance of QDA was significantly better than LDA, KDA and NN, which indicates that Gaussian density functions with different scatter matrices for different populations may not be a bad choice. Starting with this parametric set up, we got significantly lower error rates for two of the hybrid methods, method-1 and method-3. This superiority was maintained even when the same scatter matrix was used for all populations. Method-2 showed somewhat different behavior. Its performance was much better when we used the same scatter matrix instead of different scatter matrices for different populations.

In cases of crab data and wine data, in the training samples, there were very few observations from competing classes as compared to the dimensionality of the problem. LDA, since it requires fewer number of parameters to be estimated had some advantages in these cases. In both these cases, it yielded the best error rate among the parametric and nonparametric classifiers. The performance of each of hybrid method-1 and method-3 was comparable to that of LDA or QDA depending on the initial parametric set up. Method-2 had significantly lower error rates than the other two hybrid methods in the case of crab data and also in the case of wine data when different scatter matrices were used for different populations. But

when the same scatter matrix was used, it had higher error rates than those of the other two methods.

Table 4 : Average test set misclassification rates (in percentage) and their standard errors of different classifiers on benchmark data sets

	Salmon	Biomed	Iris	Diabetes	Crab	Wine	Synthetic	Satimage
LDA	7.96 (0.19)	15.48 (0.17)	2.42 (0.09)	11.04 (0.21)	5.36 (0.12)	2.21 (0.09)	10.80 (0.98)	18.03 (0.86)
QDA	7.54 (0.19)	12.51 (0.17)	2.69 (0.10)	10.38 (0.20)	6.45 (0.12)	3.24 (0.15)	10.20 (0.96)	14.68 (0.79)
KDA	8.45 (0.20)	16.77 (0.19)	2.74 (0.10)	11.93 (0.21)	6.10 (0.13)	2.33 (0.09)	11.00 (0.99)	14.33 (0.78)
NN	8.74 (0.20)	17.29 (0.20)	2.92 (0.11)	10.85 (0.20)	6.20 (0.14)	2.42 (0.10)	11.70 (1.02)	14.42 (0.79)
Hybridization based on kernels : $f_j^0 = N(\mu_j, \mathbf{\Sigma})$								
Method-1	7.80 (0.19)	15.59 (0.18)	2.42 (0.09)	11.00 (0.20)	5.35 (0.12)	2.26 (0.09)	10.60 (0.97)	14.60 (0.79)
Method-2	7.80 (0.19)	13.97 (0.21)	2.69 (0.10)	10.12 (0.18)	4.98 (0.12)	2.78 (0.11)	9.80 (0.94)	14.30 (0.78)
Method-3	7.82 (0.19)	15.73 (0.18)	2.46 (0.09)	10.92 (0.20)	5.44 (0.12)	2.21 (0.09)	10.20 (0.96)	14.24 (0.78)
Hybridization based on kernels : $f_j^0 = N(\mu_j, \mathbf{\Sigma}_j)$								
Method-1	7.50 (0.20)	12.38 (0.17)	2.68 (0.10)	10.30 (0.20)	6.42 (0.12)	3.25 (0.15)	10.10 (0.95)	14.49 (0.79)
Method-2	7.62 (0.20)	11.56 (0.19)	2.36 (0.10)	12.65 (0.22)	5.65 (0.12)	2.45 (0.09)	10.30 (0.96)	14.61 (0.79)
Method-3	7.60 (0.20)	12.06 (0.16)	2.71 (0.10)	10.36 (0.20)	6.52 (0.12)	3.24 (0.15)	10.10 (0.95)	14.37 (0.78)
Hybridization based on nearest neighbors : $f_j^0 = N(\mu_j, \mathbf{\Sigma})$								
Method-1	7.78 (0.19)	16.42 (0.19)	2.45 (0.10)	10.76 (0.20)	5.41 (0.13)	2.22 (0.09)	10.60 (0.97)	16.40 (0.83)
Method-2	7.82 (0.19)	14.75 (0.19)	2.56 (0.10)	10.30 (0.18)	4.96 (0.12)	2.54 (0.10)	10.70 (0.98)	14.54 (0.79)
Method-3	7.75 (0.19)	16.20 (0.18)	2.46 (0.10)	10.77 (0.20)	5.42 (0.12)	2.22 (0.09)	10.50 (0.97)	18.03 (0.86)
Hybridization based on nearest neighbors : $f_j^0 = N(\mu_j, \mathbf{\Sigma}_j)$								
Method-1	7.64 (0.21)	11.67 (0.16)	2.73 (0.10)	10.47 (0.20)	6.48 (0.13)	3.18 (0.15)	10.40 (0.97)	14.50 (0.79)
Method-2	7.71 (0.19)	13.53 (0.19)	2.25 (0.09)	13.50 (0.23)	5.70 (0.13)	2.60 (0.13)	10.80 (0.98)	14.25 (0.78)
Method-3	7.62 (0.20)	11.71 (0.16)	2.73 (0.10)	10.42 (0.20)	6.46 (0.12)	3.19 (0.15)	10.40 (0.97)	14.44 (0.79)

Synthetic data and satimage data are the two datasets which have specific training and test samples. On the synthetic data, nonparametric methods had slightly higher error rates as compared to parametric and hybrid classifiers. On the satimage data, all classifiers except LDA had similar error rates. Though LDA led to higher misclassification rates in this data set, most of the hybrid methods, especially the kernel based methods, could improve the classification performance inspite of starting with the same parametric set up.

5 Concluding remarks

This paper presents some hybrid classification techniques based on kernel and nearest neighbor methods. Traditional parametric classifiers often lead to poor performance when one or more model assumptions get violated. In practice, it is difficult to verify the validity of model assumptions, and improper models may lead to poor performance by the resulting classifier. One way to avoid this problem is to use nonparametric methods, which are more flexible and free from parametric model assumptions. But in addition to statistical instability, one major limitation of these nonparametric methods is their inability to incorporate the additional information that one may have on population distributions, and which can be modeled parametrically. As a result, in such cases, nonparametric methods may have higher error rates than parametric classifiers . Hybrid classifiers are designed to overcome these limitations. When the true class densities are close to the assumed parametric models, hybrid methods perform much better than their nonparametric counter parts. But unlike parametric classifiers, hybrid classifiers are not too sensitive to the choice of the initial parametric models. So, when one has some doubt about the validity of model assumptions, it is

always safe to use hybrid classifiers, which provide an automatic safeguard against possible deviations from parametric model assumptions. When the true population distributions are far from the assumed parametric models, hybrid classifiers can substantially improve the performance of parametric methods and yield misclassification rates, which are comparable to those of nonparametric classifiers. Using several simulated and real datasets, we have amply demonstrated these important features of hybrid classifiers in this article.

The multi-scale methodology discussed in Section 2.3 allows us the flexibility to consider the results for different levels of smoothing. Following the line of argument adopted for the proof of Theorem 2.1, one can also prove the convergence of the error rate for the aggregated classifiers discussed in [15] and [17]. Shalak [38] suggested to combine the classifiers when they have diversity among themselves. Hybrid classifiers with different scales of smoothing are expected to have reasonable diversity among themselves. While classifiers with large smoothing parameters behave like parametric classifiers and look at global features, classifiers with small smoothing parameters behave like nonparametric classifiers and concentrate more on the local patterns. So, aggregation of hybrid classifiers are expected to perform well, and we have observed that in our data analysis.

Appendix

Here, we will present some large sample results on the convergence of the hybrid density estimates and the proof of Theorem 1. For this asymptotic analysis, we will assume that all population density functions are smooth and have derivatives up to sufficient order, and K

is symmetric about $\mathbf{0}$ with $\int \|\mathbf{t}\|^2 K(\mathbf{t})d\mathbf{t} < \infty$ and $\int K^2(\mathbf{t})d\mathbf{t} < \infty$. Note that the Gaussian and most of the other popular kernel functions satisfy these properties. We will also assume that each of the initial parametric models f_j^0 $(j = 1, 2, \ldots, J)$ has support over the entire measurement space, and for all $j = 1, 2, \ldots, J$, n_j/n converges to the prior probability π_j $(0 < \pi_j < 1)$ as $n \to \infty$. This implies that for all $j = 1, 2, \ldots, J$, n_j/n remains bounded away from 0 and 1 as $n \to \infty$. In other words, each of the n_js are of the same asymptotic order and $n \to \infty$ implies $\min\{n_1, n_2, \ldots, n_j\} \to \infty$.

Proposition 1 : Suppose that for all $j = 1, 2, \ldots, J$, $f_j^0(\mathbf{x}, \boldsymbol{\beta}_j)$ is continuous in $\mathbf{x}$ and $\boldsymbol{\beta}_j$, and as $n \to \infty$ (which implies $n_j \to \infty$ for all $j = 1, 2, \ldots, J$), $\hat{\boldsymbol{\beta}}_j \xrightarrow{P} \boldsymbol{\beta}_j^0$ for some $\boldsymbol{\beta}_j^0$ in the parameter space irrespective of whether the parametric model is correct or not. $f_j^0(\mathbf{x}, \boldsymbol{\beta}_j^0)$ may be viewed as the best parametric approximation of f_j in that class. Also, assume that for all $j = 1, 2, \ldots, J$, $h_j \to 0$ and $n_j h_j^d \to \infty$ as $n \to \infty$. Then, $\hat{f}_{jh_j}^{(2)}(\mathbf{x})$ and $\hat{f}_{jh_j}^{(3)}(\mathbf{x})$ both converge to $f_j(\mathbf{x})$ (*in probability*) as $\min\{n_1, n_2, \ldots, n_J\} \to \infty$. Further, if K has bounded variation, and the f_j's are uniformly continuous, the above convergence result holds also for $\hat{f}_{jh_j}^{(1)}$ under a slightly stronger condition, namely $n_j h_j^d / log(n) \to \infty$ as $n \to \infty$ for $j = 1, 2, \ldots, J$.

Proof : We first consider the case of $\hat{f}_{jh_j}^{(1)}(\mathbf{x})$. To show its consistency, let us express it in the following form

$$\hat{f}_{jh_j}^{(1)}(\mathbf{x}) = \frac{\{V_{3h_j}(\mathbf{x})\}^{-1} \int_{\mathbf{y} \in B(\mathbf{x}, 3h_j)} \hat{f}_{jh_j}(\mathbf{y}) d\mathbf{y}}{\{V_{3h_j}(\mathbf{x})\}^{-1} \int_{\mathbf{y} \in B(\mathbf{x}, 3h_j)} \{f_j^0(\mathbf{y}, \hat{\boldsymbol{\beta}}) / f_j^0(\mathbf{x}, \hat{\boldsymbol{\beta}})\} d\mathbf{y}},$$

where $V_{3h_j}(\mathbf{x})$ is the volume of $B(\mathbf{x}, 3h_j)$. Since $f_j^0(\mathbf{x}, \hat{\boldsymbol{\beta}}_j)$ is continuous in $\hat{\boldsymbol{\beta}}_j$ and $\mathbf{x}$, and $\hat{\boldsymbol{\beta}}_j$ converges to $\boldsymbol{\beta}_j^0$, it is easy to show that the denominator converges to 1 *in probability*.

Now, under the assumed conditions, due to uniform convergence of kernel density estimates [39], for every $\epsilon > 0$, it is possible to find a positive integer m_1 such that for all $n \geq m_1$, $\sup_{\mathbf{y}} |\hat{f}^1_{jh_j}(\mathbf{y}) - f_j(\mathbf{y})| < \epsilon/2$. Again, because of the the continuity of f_j, one can choose another positive integer m_2 such that for all $n \geq m_2$, $\|\mathbf{x}-\mathbf{y}\| < 3h_j \Rightarrow |f_j(\mathbf{x}) - f_j(\mathbf{y})| < \epsilon/2$. This implies

$$\{V_{3h_j}(\mathbf{x})\}^{-1} \int_{\mathbf{y} \in B(\mathbf{x}, 3h_j)} |\hat{f}^1_{jh_j}(\mathbf{y}) - f_j(\mathbf{x})| \, d\mathbf{y} < \epsilon \;\text{ for all } n > max\{m_1, m_2\}.$$

So, the numerator of $\hat{f}^{(1)}_{jh_j}(\mathbf{x})$ (and hence $\hat{f}^{(1)}_{jh_j}(\mathbf{x})$ itself) converges to $f_j(\mathbf{x})$ *in probability.*

Next, we consider the case of $\hat{f}^{(2)}_{jh_j}(\mathbf{x})$. Under the assumed conditions, its consistency of follows from [22].

Finally, we consider the case of $\hat{f}^{(3)}_{jh_j}(\mathbf{x})$. Note that this density estimate can be expressed in the following form

$$\hat{f}^{(3)}_{jh_j}(\mathbf{x}) = f^0_j(\mathbf{x}, \hat{\boldsymbol{\beta}}_j) \frac{\hat{f}_{jh_j}(\mathbf{x})}{E_{\hat{f}^0_j}\{\hat{f}_{jh_j}(\mathbf{x})\}}, \quad \text{where } \hat{f}^0_j(\mathbf{x}) = f^0_j(\mathbf{x}, \hat{\boldsymbol{\beta}}_j).$$

Now, using the result on the expectation of a kernel density estimate [40], we get $E_{\hat{f}^0_j}\{\hat{f}_{jh_j}(\mathbf{x})\} = f^0_j(\mathbf{x}, \hat{\boldsymbol{\beta}}_j) + O(h_j^2)$. Since $\hat{\boldsymbol{\beta}}_j$ converges to $\boldsymbol{\beta}^0_j$, and h_j converges to 0 as n_j tends to ∞, it is easy to show that $f^0_j(\mathbf{x}, \hat{\boldsymbol{\beta}}_j)$ and $E_{\hat{f}^0_j}\{\hat{f}_{jh_j}(\mathbf{x})\}$ both tend to $f^0_j(\mathbf{x}, \boldsymbol{\beta}^0_j)$ as $n_j \to \infty$, and their ratio converges to 1. Now, under assumed conditions, the convergence of $\hat{f}^{(3)}_{jh_j}(\mathbf{x})$ follows from the convergence of the kernel density estimate $\hat{f}_{jh_j}(\mathbf{x})$. □

Lemma 1 : Suppose that $\mathbf{x}_{j1}, \mathbf{x}_{j2}, \ldots, \mathbf{x}_{jn_j}$ $(\sum n_j = n)$ are independent observations from a continuous density function f_j $(j = 1, 2, \ldots, J)$. Consider a data point $\mathbf{x}$ and define $\mathbf{x}^{(n,k)}, B_{n,k}(\mathbf{x})$ and $V_{n,k}(\mathbf{x})$ as in Section 2.2. Now, assume that $k \to \infty, k/n \to 0$ and also

recall that $n_j/n \to \pi_j \ \ (0 < \pi_j < 1, \sum \pi_j = 1)$ as $n \to \infty$. Then, for all $j = 1, 2, \ldots, J$, $k_j/n_j V_{n,k}(\mathbf{x}) \xrightarrow{P} f_j(\mathbf{x})$ as $n \to \infty$.

Proof : Write $k_j/n_j V_{n,k}(\mathbf{x})$ as $\{k_j/k\} \times \{n/n_j\} \times \{k/nV_{n,k}(\mathbf{x})\}$. Now, under assumed conditions, k_j/k converges (*in probability*) to the posterior probability $\pi_j f_j(\mathbf{x}) / \sum \pi_t f_t(\mathbf{x})$ [6]. According to our condition, n/n_j converges to $1/\pi_j$. From the consistency of nearest neighbor density estimates [30], we get the convergence (*in probability*) of $k/nV_{n,k}(\mathbf{x})$ to $\sum \pi_t f_t(\mathbf{x})$. Hence, $k_j/n_j V_{n,k}(\mathbf{x}) \xrightarrow{P} f_j(\mathbf{x})$ for all $j = 1, 2, \ldots, J$ as the sample size $n \to \infty$. □

Proposition 2 : Suppose that for all $j = 1, 2, \ldots, J$, $f_j^0(\mathbf{x}, \boldsymbol{\beta}_j)$ is continuous in $\mathbf{x}$ and $\boldsymbol{\beta}_j$. Define $\boldsymbol{\beta}_j^0$ and $\hat{\boldsymbol{\beta}}_j$ as in Proposition 1. Also, assume that as the training sample size $n \to \infty$, $k \to \infty$, $k/n \to 0$ and $\hat{\boldsymbol{\beta}}_j \xrightarrow{P} \boldsymbol{\beta}_j^0$ for all $j = 1, 2, \ldots, J$. Then, each of the three hybrid density estimates $\hat{f}_{j,k}^{(1)}(\mathbf{x})$, $\hat{f}_{j,k}^{(2)}(\mathbf{x})$ and $\hat{f}_{j,k}^{(3)}(\mathbf{x})$ asymptotically converges (*in probability*) to $f_j(\mathbf{x})$.

Proof : We first consider the case of $\hat{f}_{j,k}^{(1)}(\mathbf{x})$, which can be expressed as

$$\hat{f}_{j,k}^{(1)}(\mathbf{x}) = \frac{k_j/n_j V_{n,k}(\mathbf{x})}{\{V_{n,k}(\mathbf{x})\}^{-1} \int_{\mathbf{y} \in B_{n,k}(\mathbf{x})} \{f_j(\mathbf{y}, \hat{\boldsymbol{\beta}}_j)/f_j(\mathbf{x}, \hat{\boldsymbol{\beta}}_j)\} d\mathbf{y}}.$$

Now, note that under assumed conditions, as $n \to \infty$, the radius of $B_{n,k}(\mathbf{x})$ shrinks to 0. Therefore, using the continuity of f_j^0 and the convergence of $\hat{\boldsymbol{\beta}}_j$, it is easy to show that the denominator converges to 1. Now, Lemma 1 suggests that the numerator $\hat{f}_{j,k}^{(1)}(\mathbf{x})$ and hence $\hat{f}_{j,k}^{(1)}(\mathbf{x})$ converges to $f_j(\mathbf{x})$ *in probability*.

Next, we consider the case of $f_{j,k}^{(2)}(\mathbf{x})$. Note that this can be expressed as

$$\hat{f}_{j,k}^{(2)}(\mathbf{x}) = \{k_j/n_j V_{n,k}(\mathbf{x})\} \times \left[\frac{1}{k_j} \sum_{x_{ji}: x_{ji} \in B_{n,k}(\mathbf{x})} \{f_j^0(\mathbf{x}, \hat{\boldsymbol{\beta}}_j)/f_j^0(\mathbf{x}_{ji}, \hat{\boldsymbol{\beta}}_j)\} \right]$$

Since the radius of $B_{n,k}(\mathbf{x})$ shrinks to 0 as $n \to \infty$, using the continuity of f_j^0 and the

convergence of $\hat{\boldsymbol{\beta}}_j$, for every $\epsilon > 0$, it is possible to choose n_0 such that for all $n > n_0$, $\mathbf{y} \in B_{n,k}(\mathbf{x}) \Rightarrow 1-\epsilon < \{f_j^0(\mathbf{x}, \hat{\boldsymbol{\beta}}_j)/f_j^0(\mathbf{x}_{ji}, \hat{\boldsymbol{\beta}}_j)\} < 1+\epsilon$. Therefore, the second term of $\hat{f}_{j,k}^{(2)}(\mathbf{x})$ converges to 1. Now, the convergence of the first term and hence that of $\hat{f}_{j,k}^{(2)}(\mathbf{x})$ follows from Lemma 1.

Finally, we consider the case of $f_{j,k}^{(3)}(\mathbf{x})$. It follows from Lemma 1 that $k_j/n_j V_{n,k}(\mathbf{x})$ is a consistent estimate of the population density function $f_j(\mathbf{x})$. Therefore, $E_{\hat{f}^0}\left(k_j/n_j V_{n,k}(\mathbf{x})\right)$ has the expression of the form $\hat{f}_j^0(\mathbf{x}) + r_n$, where r_n converges to 0 and $\hat{f}_j^0(\mathbf{x}) = f_j^0(\mathbf{x}, \hat{\boldsymbol{\beta}}_j)$ converges to $f_j^0(\mathbf{x}, \boldsymbol{\beta}_j^0)$ *in probability* as $n \to \infty$ (using the continuity of f_j^0). In particular, $E_{\hat{f}^0}\left(k_j/n_j V_{n,k}(\mathbf{x})\right) = \hat{f}_j^0(\mathbf{x}) + O(k/n)^2$ [28], [31] for the asymptotic mean of the nearest neighbor density estimate. Therefore, $\hat{f}_j^0(\mathbf{x})/E_{\hat{f}^0}\{k_j/n_j V_{n,k}(\mathbf{x})\}$ converges to 1, and the result follows from Lemma 1. □

Proof of Theorem 1 : Let us define $T_{nj}(\mathbf{x}) = \sum_{L_n}^{U_n} W(s)\lambda_{nj}^s(\mathbf{x})$ where $\lambda_{nj}^s = \pi_j f_{js}^*(\mathbf{x})/\sum_t \pi_t f_{ts}^*(\mathbf{x})$. It is quite transparent from Propositions 1 and 2 that under the assumed conditions on L_n and U_n, for any sequence of smoothing parameters in the interval $[L_n, U_n]$, $\lambda_{nj}^s(\mathbf{x})$ converges (*in probability*) to the posterior probability $p(j \mid \mathbf{x}) = \pi_j f_j(\mathbf{x})/\sum_t \pi_t f_t(\mathbf{x})$ as $n \to \infty$. If we can show similar convergence for $T_{n,j}$, the proof can be completed using the Dominated Convergence Theorem and following the argument given in the first paragraph of section 2.4.

If possible, suppose that $T_{nj}(\mathbf{x})$ does not converge to $p(j \mid \mathbf{x})$ as $n \to \infty$. Then, there exist an $\epsilon > 0$ and a subsequence $\{T_{n'j}; n' \geq 1\}$ such that $|T_{n'j}(\mathbf{x}) - p(j \mid \mathbf{x})| > \epsilon$ for all $n' \geq 1$. Since $T_{n'j}$ is a weighted average of $\lambda_{n'j}^s(\mathbf{x})$s, one can always find some $s = s(n')$ $(L_{n'} \leq s(n') \leq U_{n'})$ such that $|\lambda_{n'j}^{s(n')}(\mathbf{x}) - p(j \mid \mathbf{x})| > \epsilon$. So, we can construct a sequence of smoothing parameters $\{s(n'); n' \geq 1\}$ such that along that sequence $|\lambda_{n'j}^{s(n')}(\mathbf{x}) - p(j \mid \mathbf{x})| > \epsilon$

for all $n' \geq 1$. But this sequence of smoothing parameters satisfies the conditions required for the consistency of the hybrid density estimates, and hence $\lambda_{n'j}^{s(n')}(\mathbf{x})$ should converge to $p(j \mid \mathbf{x})$ *in probability*. This is a contradiction. □

References

[1] Anderson, T. W. (1984). *An Introduction to Multivariate Statistical Analysis*. Wiley, New York.

[2] Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984) *Classification and Regression Trees*. Wadsworth and Brooks Press, Monterrey, California.

[3] Breiman, L. (1996) Bagging predictors. *Machine Learning*, **24**, 123-140.

[4] Chaudhuri, P. and Marron, J. S. (1999). SiZer for exploration of structures in curves. *J. Amer. Statist. Assoc.* **94**, 807-823.

[5] Chaudhuri, P. and Marron, J. S. (2000). Scale space view of curve estimation. *Ann. Statist.* **28**, 408-428.

[6] Cover, T. M. and Hart, P. E. (1967) Nearest neighbor pattern classification, *IEEE Trans. Info. Theory*, **13**, 21-27.

[7] Dasarathy, B. V. (1991) *Nearest Neighbor (NN) Norms : NN Pattern Classification Techniques*. IEEE Computer Society, Washington.

[8] Donoho, D., Johnstone, I., Karkyacharian, G. and Picard, D. (1996) Density estimation by wavelet thresholding. *Ann. Statist.*, **24**, 508-539.

[9] Duda, R., Hart, P. and Stork, D. G. (2000) *Pattern Classification.* Wiley, New York.

[10] Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, **7**, 179-188.

[11] Fix, E. and Hodges, J. L., Jr.,(1951) Discriminatory analysis, nonparametric discrimination, consistency properties. *Randolph Field, Texas, Project 21-49-004, Report No. 4.*

[12] Friedman, J. H. (1994) Flexible metric nearest neighbor classification. *Technical Report, Department of Statistics, Stanford University.*

[13] Friedman, J. H., Hastie, T. and Tibshirani, R. (2000) Additive logistic regression : a statistical view of boosting (with discussion). *Ann. Statist.*, **28**, 337-374.

[14] Fukunaga, K. and Hostetler, L. D. (1973) Optimization of k-nearest neighbor density estimates. *IEEE Trans. Info. Theory*, **19**, 320-326.

[15] Ghosh, A. K., Chaudhuri, P. and Murthy, C. A. (2005) On visualization and aggregation of nearest neighbor classifiers. *IEEE Trans. Pattern Anal. Machine Intell.*, **27**, 1592-1602.

[16] Ghosh, A. K., Chaudhuri, P. and Murthy, C. A. (2006) Multi-scale classification using nearest neighbor density estimates. *IEEE Trans. Sys. Man, Cybern., Part B*, **36**, 1139-1148.

[17] Ghosh, A. K., Chaudhuri, P. and Sengupta, D. (2006) Classification using kernel density estimates : multi-scale analysis and visualization. *Technometrics*, **48**, 120-132.

[18] Godtliebsen, F., Marron, J. S. and Chaudhuri, P. (2002) Significance in scale space for bivariate density estimation. *J. Comput. Graphical Statist.*, **11**, 1-22.

[19] Hand, D. J. (1982) *Kernel Discriminant Analysis.* Wiley, Chichester.

[20] Hastie, T. and Tibshirani, R. (1996) Discriminant adaptive nearest neighbor classification. *IEEE Trans. Pattern Anal. Machine Intell.*, **18**, 607-616.

[21] Hastie, T., Tibshirani, R. and Friedman, J. H. (2001) *The Elements of Statistical Learning : Data Mining, Inference and Prediction*. Springer Verlag.

[22] Hjort, N. L. and Glad, I. (1995) Nonparametric density estimation with a parametric start. *Ann. Statist.*, **23**, 882-904.

[23] Hjort, N. L. and Jones, M. C. (1996) Locally parametric nonparametric density estimation. *Ann. Statist.*, **24**, 1619-1647.

[24] Hoti, F. and Holmstrom, L. (2004) A semiparametric density estimation approach to pattern classification. *Pattern Recognition*, **37**, 409-419.

[25] Johnson, R. A. and Wichern, D. W. (1992) *Applied Multivariate Statistical Analysis*. Prentice Hall, New Jersey.

[26] Jones, M. C., Linton, O. and Neilsen, J. P. (1995) A simple and effective bias reduction method for density and regression estimation. *Biometrika*, **82**, 327-338.

[27] Kooperberg, C. and Stone, C. J. (1991) A study of logspline density estimation. *Comput. Statist. Data Anal.*, **7**, 327-347.

[28] Lai, S. L. (1977) Large sample properties of k-nearest neighbor procedures. *Ph.D. Dissertation, Dept. of Mathematics, UCLA, Los Angeles*.

[29] Lachenbruch, P. A. and Mickey, M. R. (1968) Estimation of error rates in discriminant analysis. *Technometrics*, **10**, 1-11.

[30] Loftsgaarden, D. O. and Quesenberry, C. P. (1965) A nonparametric estimate of a multivariate density function. *Ann. Math. Statist.*, **36**, 1049-1051.

[31] Mack, Y. P. (1981) Local properties of k-nn regression estimates. *SIAM J. Alg. Disc. Math.*, **2**, 311-323.

[32] Mahalanobis, P. C. (1936) On the generalized distance in statistics. *Proc. Nat. Acad. Sci., India*, **12**, 49-55.

[33] McLachlan. G. J. (1992) *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York.

[34] Olkin, I. and Spiegelman, C. H. (1987) A Semiparametric approach to density estimation. *J. Amer. Statist. Assoc.*, **82**, 858-865.

[35] Ripley, B. D. (1996) *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge.

[36] Schapire, R. E., Fruend, Y., Bartlett, P. and Lee, W. (1998) Boosting the margin : a new explanation for the effectiveness of voting methods. *Ann. Statist.*, **26**, 1651-1686.

[37] Scott, D. W. (1992) *Multivariate Density Estimation : Theory, Practice and Visualization*. Wiley, New York.

[38] Shalak, D. B. (1996) Prototype selections for composite nearest neighbor classifiers. *Ph.D. Thesis, Department of Computer Science, University of Massachusetts*.

[39] Silverman, B. W. (1978) Weak and strong uniform consistency of the kernel estimate of a density function and its derivatives. *Ann. Statist.*, **6**, 177-184.

[40] Silverman, B. W. (1986) *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London.

[41] Stone, M. (1977) Cross validation : a review. *Mathematische Operationsforschung und Statistik, Series Statistics*, **9**, 127-139.

[42] Vapnik. V. N. (1998) *Statistical Learning Theory*. Wiley, New York.

[43] Wand, M. and Jones, M. C. (1995) *Kernel Smoothing*, Chapman and Hall, London.