Correspondence

Kernel Discriminant Analysis Using Case-Specific Smoothing Parameters

Anil K. Ghosh

***Abstract*—In kernel discriminant analysis, one common practice is to use a fixed level of smoothing (estimated from training data) for classifying all unlabeled observations. But, in classification, a good choice of smoothing parameters also depends on the observation to be classified. Therefore, instead of using a fixed level of smoothing over the entire measurement space, it may be more useful to estimate the smoothing parameters depending on that specific observation. Here, we propose a simple method for this case-specific smoothing. Some benchmark data sets are analyzed to illustrate the performance of the proposed method.**

***Index Terms*—Bandwidth, Bayes risk, bootstrap, cross validation, kernel smoothing, misclassification rate, nearest neighbor, p-value.**

I. Introduction

Kernel density estimation [26], [28] is a popular method for nonparametric density estimation, and it has one well-known application in kernel discriminant analysis (KDA) [13]. In a J-class classification problem, if we have a training sample $\mathcal{S} = \{(\mathbf{x}_i, c_i); \mathbf{x}_i \in R^d, C_i \in \{1, 2, \ldots, J\}, i = 1, 2, \ldots, n\}$ of n observations, the kernel estimate for the density function $f_j (j = 1, 2, \ldots, J)$ can be expressed as $\hat{f}_{jh}(\mathbf{x}) = n_j^{-1} h^{-d} \sum_{i:c_i=j} K\{h^{-1}(\mathbf{x} - \mathbf{x}_i)\}$, where n_j is the number of observations from the jth class ($\sum n_j = n$), K is a d-dimensional density function symmetric around $\mathbf{0}$, and h is the associated smoothing parameter known as the bandwidth. These kernel density estimates are used to construct the KDA rule given by $\delta_h(\mathbf{x}) = \arg\max \pi_j \hat{f}_{jh}(\mathbf{x})$, where π_j is the prior probability of the jth class. If these priors are not known, one usually estimates them using training sample proportions $\hat{\pi}_j = n_j/n (j = 1, 2, \ldots, J)$ of different classes. Many choices for the kernel function K are available in the literature [26], [28]. Here, we will assume $K(\mathbf{t}) = (2\pi)^{-d/2} \exp(-\|\mathbf{t}\|^2/2)$ and all competing density functions are continuous.

For $J = 2$, we can look at KDA also from regression perspective. Like many other classification techniques, here, we can define an indicator variable Y, which takes the value 1 and -1 when the observation comes from Class-1 and Class-2, respectively. Under this set up, for any given $\mathbf{x}$, $g(\mathbf{x}) = \mathbf{E}(\mathbf{Y}|\mathbf{x}) = \mathbf{p}(\mathbf{1}|\mathbf{x}) - \mathbf{p}(\mathbf{2}|\mathbf{x})$ gives the difference between the true posterior probabilities of the two classes. If we use the kernel method for regression, the Nadaraya–Watson [34] estimate for Y is given by

$$\hat{Y} = \sum_{i=1}^{n} K\left\{h^{-1}(\mathbf{x} - \mathbf{x}_i)\right\} Y_i \Big/ \sum_{i=1}^{n} K\left\{h^{-1}(\mathbf{x} - \mathbf{x}_i)\right\}$$
$$= \frac{\hat{\pi}_1 \hat{f}_{1h}(\mathbf{x}) - \hat{\pi}_2 \hat{f}_{2h}(\mathbf{x})}{\hat{\pi}_1 \hat{f}_{1h}(\mathbf{x}) + \hat{\pi}_2 \hat{f}_{2h}(\mathbf{x})} = \hat{p}(1|\mathbf{x}) - \hat{p}(2|\mathbf{x}).$$

So, one can classify the observation $\mathbf{x}$ depending on the sign of $\hat{Y}$.

Manuscript received August 30, 2007; revised January 26, 2008. This paper was recommended by Associate Editor P. Sastry.

The author was with the Department of Mathematics and Statistics, Indian Institute of Technology, Kanpur 208016, India. He is now with the Theoretical Statistics and Mathematics Unit, Indian Statistical Institute, Kolkata 700108, India (e-mail: akghosh@isical.ac.in).

Digital Object Identifier 10.1109/TSMCB.2008.925754

The smoothing parameter h plays an important role in both regression and classification problems. Since KDA is derived using kernel estimates of population densities, one may be tempted to use the value of h that is optimum for kernel density estimation. Several algorithms are available in the literature [17], [27], [34] for this purpose. However, while those techniques are quite good for giving low mean integrated squared error (MISE) of the density estimate, they may not be appropriate for classification problems [7], [29], [35]. Looking at the classification problem from regression perspective, one can also adopt the bandwidth selection algorithms [14], [19], [22], [34] available for kernel regression. These algorithms, in some sense, have the same spirit as [11], where the authors proposed to choose the bandwidth by minimizing the MISE of the density difference. However, in classification, instead of global accuracy of the regression surface, we are mainly interested in the regression surface near $g(\mathbf{x}) = 0$. Therefore, the bandwidths obtained by minimization of global errors may not work well. In classification problems, one usually minimizes the cross validation [23] or the bootstrap [5], [12] estimate of the error rate to select the optimum value of h. Other bandwidth selection algorithms for classification problems are available in [7] and [12].

All these algorithms use the training data to select the optimum bandwidth, which is then used for classification of all observations. However, one should note that in addition to depending on the training data, a good choice of h depends on the observation to be classified. Therefore, instead of using a fixed h over the entire measurement space, it may be more useful to choose the bandwidth adaptively depending on that specific observation. The importance of this adaptive smoothing has been investigated in the context of density estimation [1], [24], [31] and regression [20], [25], [30], [33]. However, those algorithms were designed to achieve different goals, and they may not be useful for classification, where instead of accuracy of the density estimate (or regression estimate), the main emphasis is on the accuracy of the decision rule. In practice, it is possible to have poor estimates of density functions (or regression functions) which lead to a good classifier [6], [7]. Therefore, in classification, one needs to develop a different method for adaptive bandwidth selection. In this correspondence paper, we propose one such adaptive bandwidth selection algorithm for KDA, where the bandwidth h is chosen not only depending on the training data, but also based on the observation to be classified. The use of the same bandwidth h in all directions requires some preliminary transformation (standardization) of the data. We will use the usual moment-based estimate of the dispersion matrix for this purpose.

II. Case-Specific Choice of h

For the case-specific choice of h, we associate the two-class classification problem (or the regression problem) with the problem of statistical hypothesis testing. Given the query point $\mathbf{x}$, for any h, one observes either $\hat{Y} > 0$ or $\hat{Y} < 0$, but from a statistical point of view, it is important to know whether the difference between $\hat{Y}$ and 0 is statistically significant. Or, in other words, one may be interested in knowing whether there is any significant difference between $\hat{\pi}_1 \hat{f}_{1h}(\mathbf{x})$ and $\hat{\pi}_2 \hat{f}_{2h}(\mathbf{x})$ in terms of their expected values. This leads to the hypothesis testing problem $H_0 : E\{\hat{\pi}_1 \hat{f}_{1h}(\mathbf{x})\} \geq E\{\hat{\pi}_2 \hat{f}_{2h}(\mathbf{x})\}$ against $H_1 : E\{\hat{\pi}_1 \hat{f}_{1h}(\mathbf{x})\} < E\{\hat{\pi}_2 \hat{f}_{2h}(\mathbf{x})\}$. Kernel density estimates are averages of independent identically distributed (i.i.d.) random variables.

Therefore, for a reasonably large sample, one can assume normality of their distributions. Since $\hat{\pi}_1 \xrightarrow{P} \pi_1$ and $\hat{\pi}_2 \xrightarrow{P} \pi_2$, from Slutskey's lemma, it follows that $[\{\hat{\pi}_1 \hat{f}_{1h}(\mathbf{x}) - \hat{\pi}_2 \hat{f}_{2h}(\mathbf{x})\} - E\{\hat{\pi}_1 \hat{f}_{1h}(\mathbf{x}) - \hat{\pi}_2 \hat{f}_{2h}(\mathbf{x})\}]/\sqrt{\pi_1^2 s_{1h}^2(\mathbf{x}) + \pi_2^2 s_{2h}^2(\mathbf{x})} \xrightarrow{L} N(0,1)$, where $s_{ih}^2(\mathbf{x})$ is the variance of $\hat{f}_{ih}(\mathbf{x})$ $(i = 1, 2)$. Note that $s_{ih}^2(\mathbf{x}) = h^{-2d}\mathrm{Var}(K\{(\mathbf{x} - \mathbf{X})/h\})/n_i$, where $\mathbf{X} \sim f_i$, and one can use the sample variance of $K\{(\mathbf{x} - \mathbf{X})/h\}$ to get its unbiased estimate $\hat{s}_{ih}^2$. Since for any fixed h, $\hat{\pi}_1^2 \hat{s}_{1h}^2(\mathbf{x}) + \hat{\pi}_2^2 \hat{s}_{2h}^2(\mathbf{x}) \xrightarrow{P} \pi_1^2 s_{1h}^2(\mathbf{x}) + \pi_2^2 s_{2h}^2(\mathbf{x})$, one can use $\Phi(\{\hat{\pi}_1 \hat{f}_{1h}(\mathbf{x}) - \hat{\pi}_2 \hat{f}_{2h}(\mathbf{x})\}/\sqrt{\hat{\pi}_1^2 \hat{s}_{1h}^2(\mathbf{x}) + \hat{\pi}_2^2 \hat{s}_{2h}^2(\mathbf{x})})$ as the one-sided p-value for this testing problem, where Φ is the cumulative distribution function of the standard normal distribution. Clearly, higher values of $\alpha_h(\mathbf{x}) = |\hat{\pi}_1 \hat{f}_{1h}(\mathbf{x}) - \hat{\pi}_2 \hat{f}_{2h}(\mathbf{x})|/\sqrt{\hat{\pi}_1^2 \hat{s}_{1h}^2(\mathbf{x}) + \hat{\pi}_2^2 \hat{s}_{2h}^2(\mathbf{x})} = |n_1 \hat{f}_{1h}(\mathbf{x}) - n_2 \hat{f}_{2h}(\mathbf{x})|/\sqrt{n_1^2 \hat{s}_{1h}^2(\mathbf{x}) + n_2^2 \hat{s}_{2h}^2(\mathbf{x})}$ lead to p-values close to 0 or 1 indicating stronger evidence in favor of one of the two classes. Therefore, one can choose h by maximizing $\alpha_h(\mathbf{x})$ over a suitable interval, and this maximizer $h_{\mathbf{x}}$ can be used to construct the adaptive decision rule $\delta^*(\mathbf{x}) = \arg\max \hat{\pi}_j \hat{f}_{jh_{\mathbf{x}}}(\mathbf{x})$. Although we have used the Gaussian kernel throughout this correspondence paper, this case-specific bandwidth selection method can be used for other kernel functions as well. Note that, in addition to training sample observations, here, we use an unlabeled observation $\mathbf{x}$ to construct the classifier, which is used to predict the class label of the observation $\mathbf{x}$ only. Therefore, in that sense, the resulting classifier can be viewed as a semisupervised or transductive classifier [3], [36].

If we denote the kth nearest neighbor of $\mathbf{x}$ by $\mathbf{x}^{(k)}$, one can notice that for $h < \|\mathbf{x} - \mathbf{x}^{(1)}\|/3$, both $\hat{f}_{1h}(\mathbf{x})$ and $\hat{f}_{2h}(\mathbf{x})$ will be very close to 0, and that will give no useful information for classification. So, it is somewhat reasonable to use $\|\mathbf{x} - \mathbf{x}^{(1)}\|/3$ as the lower limit for h (call it $h_{\mathrm{L}}(\mathbf{x})$). On the other hand, the use of very large h not only increases the computing cost by increasing the length of the search interval, but it also fails to represent the local pattern of the measurement space. Moreover, if the prior probabilities of two competing classes are not equal, the use of large h always leads to a decision in favor of the class having the larger prior [7]. Therefore, if the training samples from the two classes are not of comparable sizes, and if we estimate π_j by $\hat{\pi}_j = n_j/n$, large h will always favor the bigger class, which is not desirable. To set the upper limit of h (call it $h_{\mathrm{U}}(\mathbf{x})$), one can borrow the idea from k-nearest neighbor classification [4], [18], where we consider only k out of n neighbors of $\mathbf{x}$ to predict its class label. Note that if we use $h_{\mathrm{U}}(\mathbf{x}) = \|\mathbf{x}, \mathbf{x}^{(k)}\|/3$, only first k neighbors of $\mathbf{x}$ will have significant contributions to $\hat{f}_{1h}(\mathbf{x})$ and $\hat{f}_{2h}(\mathbf{x})$, and the resulting decision rule will be like a weighted nearest neighbor rule [2]. For nearest neighbor classification, asymptotic results [4], [18] suggest that k should tend to infinity and k/n should tend to 0 as n tends to infinity. Here, also, we can use one such choice of k to compute $h_{\mathrm{U}}(\mathbf{x})$, and for this choice of $h_{\mathrm{U}}(\mathbf{x})$, error rate of the proposed classifier δ^* converges to the optimal Bayes risk. This result is formally presented in the following theorem, and the proof is given in [10].

Theorem 1: Let $h_{\mathbf{x}}$ be the maximizer of $\alpha_h(\mathbf{x})$ in the interval $[h_{\mathrm{L}}(\mathbf{x}) = d(\mathbf{x}, \mathbf{x}^{(1)})/3, h_{\mathrm{U}}(\mathbf{x}) = d(\mathbf{x}, \mathbf{x}^{(k)})/3]$, where $k \to \infty$ and $k/n \to 0$ as $n \to \infty$. Then, the true misclassification rate of the proposed classifier $\delta^*(\mathbf{x})$ converges to the optimal Bayes risk.

This theorem suggests that $k = C\sqrt{n}$ could be a good option for some suitably chosen constant C, and here we follow [8] to take $k = 2\sqrt{n}$. We compute $\alpha_h(\mathbf{x})$ for 100 different values of h in the interval $[h_{\mathrm{L}}(\mathbf{x}), h_{\mathrm{U}}(\mathbf{x})]$ and choose the one that maximizes $\alpha_h(\mathbf{x})$.

In classification problems involving J $(J > 2)$ classes, one can perform $\binom{J}{2}$ binary classifications taking one pair of classes at a time and combine the results of all pairwise classifications by using majority voting or coupling algorithm [16]. To keep our algorithm simpler, here, we adopt the voting method, where the tied cases are arbitrarily assigned to one of the classes having the maximum number of votes.

TABLE I
BRIEF DESCRIPTION OF BENCHMARK DATA SETS

	Biomed	Glass(2C)	BUPA	Pima	Image	Vowel	Vowel-2	Diabetes	Vehicle	Glass
d	4	5	5	8	9	10	2	5	18	5
J	2	2	2	2	7	11	10	3	4	6
n	100	100	200	400	210	528	338	100	400	100
n_0	94	46	145	368	2100	462	333	45	446	114

d=dimension, J=no. of classes, n=training sample size, n_0=test sample size.

However, instead of pairwise classification, one can simultaneously deal with all competing populations and assign the observation to the class having the maximum value of $\hat{\pi}_j \hat{f}_{jh_{\mathbf{x}}}(\mathbf{x})$. To choose $h_{\mathbf{x}}$, we can set $h_{\mathrm{L}}(\mathbf{x})$ and $h_{\mathrm{U}}(\mathbf{x})$ as before, but here, we need an appropriate analog for $\alpha_h(\mathbf{x})$ to be maximized. For a pair of classes i and j, let us define $\beta_h^{(i,j)}(\mathbf{x}) = \{n_i \hat{f}_{ih}(\mathbf{x}) - n_j \hat{f}_{jh}(\mathbf{x})\}/\sqrt{n_i^2 \hat{s}_{ih}^2(\mathbf{x}) + n_j^2 \hat{s}_{ih}^2(\mathbf{x})}$ and $\beta_h^{(i)}(\mathbf{x}) = \min_{j \neq i} \beta_h^{(i,j)}(\mathbf{x})$. Note that $\beta_h^{(j,i)}(\mathbf{x}) = -\beta_h^{(i,j)}(\mathbf{x})$, and for any given $\mathbf{x}$ and h, $\beta_h^{(i)}(\mathbf{x})$ is positive only for the winning class. Also, $\alpha_h^*(\mathbf{x}) = \max_i \beta_h^{(i)}(\mathbf{x})$ is expected to be higher when we have stronger evidence in favor of the winning class. Therefore, $\alpha_h^*(\mathbf{x})$ can serve as a measure of evidence in favor of our decision, and we can maximize $\alpha_h^*(\mathbf{x})$ with respect to h to choose the bandwidth $h_{\mathbf{x}}$ to be used for classification of $\mathbf{x}$. In the following sections, for data analytic purpose, along with the pairwise approach, we will also use this method, and we will refer to them as the pairwise method and the combined method, respectively. One can verify that the multiclass version of Theorem 1 holds for both these methods.

III. RESULTS FROM THE ANALYSIS OF BENCHMARK DATA SETS

We use ten benchmark data sets to illustrate the performance of the proposed method. Four of these data sets have observations from two competing classes, whereas the rest consists of observations from three or more populations. A brief description of these data sets is given in Table I. The 2-D vowel data set (referred to as vowel-2 in Table I and Tables III–V) was generated by Petersen and Barney [21], and detail description of this data set can be obtained in that article. The rest of the data sets and their descriptions are available either at University of California, Irvine (UCI) machine learning repository (http://www.uci.ics.edu/~mlearn) or at CMU data archive (http://lib.stat.cmu.edu).

Three of these data sets, namely the image segmentation data, the vowel recognition data, and the 2-D vowel data (vowel-2), have specific training and test sets, For these data sets, test set error rates of different methods are reported in Table III. The rest of the data sets do not have specific training and test sets. Therefore, we formed these sets by randomly partitioning the data. This random partitioning was carried out 500 times to generate 500 training and test samples. For these data sets, average test set error rates (over these 500 samples) of different methods are reported in Tables II and III along with their corresponding standard errors. The sizes of the training and test samples in each partition are reported in Table I. Throughout this section, sample proportions of different classes are used as their prior probabilities, and in all these cases, we standardized the data set using the usual moment based estimate of the pooled dispersion matrix.

A. Classification Between Two Competing Populations

Here, we use four data sets, namely the biomedical data, the Pima Indian diabetes data, the BUPA liver disorder data and a subset of the glass data, to evaluate the performance of the proposed algorithm. For

TABLE II
AVERAGE MISCLASSIFICATION (IN PERCENT) RATES AND THEIR STANDARD ERRORS FOR TWO-CLASS PROBLEMS

	Biomed	Glass (2C)	BUPA	Pima
MISE-band.	17.01(.15)	20.64(.23)	34.06(.15)	29.13(.08)
Sq.root law	16.65(.14)	21.13(.23)	34.37(.15)	28.96(.08)
Regression	16.67(.13)	21.42(.24)	33.85(.15)	26.22(.07)
Adapt-reg.	17.28(.14)	22.27(.23)	33.87(.15)	28.98(.08)
Cross-valid.	16.93(.13)	22.12(.24)	33.14(.14)	26.58(.08)
Bootstrap	16.58(.13)	21.70(.24)	33.11(.14)	26.28(.07)
Opt-band.	16.78(.14)	22.39(.25)	33.38(.14)	26.05(.07)
Proposed	16.60(.13)	20.84(.24)	32.30(.14)	26.60(.07)

TABLE III
AVERAGE MISCLASSIFICATION (IN PERCENT) RATES AND THEIR STANDARD ERRORS FOR MULTICLASS PROBLEMS

	Image	Vowel	Vowel-2	Diabetes	Vehicle	Glass
MISE-band.	14.71	66.23	18.62	12.15(.19)	22.66(.08)	36.94(.19)
Sq.root law	14.76	66.45	18.32	12.22(.19)	22.65(.08)	36.19(.20)
Regression	8.38	46.54	19.52	10.63(.20)	20.32(.07)	33.52(.16)
Adapt-reg.	8.71	47.19	24.02	13.89(.19)	22.70(.08)	33.40(.16)
Cross-valid.	10.62	45.89	20.12	10.97(.19)	20.92(.11)	31.35(.17)
Bootstrap	8.00	46.75	19.52	11.10(.20)	20.99(.07)	31.01(.17)
Opt-band.	10.38	44.59	18.92	12.31(.18)	20.74(.07)	33.28(.19)
Proposed(p)	5.90	44.59	22.52	10.70(.18)	20.35(.07)	31.54(.16)
Proposed(c)	6.24	45.24	19.82	10.54(.17)	20.38(.07)	31.60(.16)

the biomedical data, we removed the observations with missing values, and carried out our analysis with the rest 194 observations. In the BUPA liver disorder data set, we did not consider the discrete variable "number of half-pint equivalents of alcohol beverages drunk per day" and used the other five variables for our analysis. In the case of glass data, although there are 214 observations from six different classes, here, we considered only 146 observations coming from two larger classes. An analysis based on all 214 observations will be reported later. In tables, we will refer to this full data set and its two-class subset as glass data and glass (2C) data, respectively. However, in this data set there are four measurement variables for which majority of the values are zero. We ignored them and used the other five variables.

For these data sets, along with the performance of our proposed method, we also report the error rates for other standard methods of bandwidth selection. In density estimation problems, one usually adopts the least squared cross-validation [26], [28] method to find out the bandwidths by minimizing the estimated MISE of the density estimates. We have used this method here to find out the optimal bandwidths and the resulting classifier (referred to as the MISE-bandwidth classifier in Tables II, III, and V). Instead of using a fixed bandwidth for density estimation, one can also go for adaptive kernel density estimation [1], [24], [31]. Because of the simplicity and the ease of implementation, here, we have used Abramson's [1] square root law for this purpose. In regression problems, one common practice is to use the least squared cross-validation method to find the optimal bandwidth, and here also, we have used that method for bandwidth selection and classification. For adaptive regression using optimal local bandwidths, we followed the databased bandwidth selection algorithm of Vieu [33], where 0.05 times the maximum pairwise distance between the measurement vectors was used as the bandwidth for the weight function. In addition to these four methods, we have also reported the performance of the cross-validation method [23] and that of the bootstrap method proposed in [5]. Because of the stepwise nature of the estimated error rate, cross-validation method often leads to multiple minimizers of the error rate. Following the suggestion of [7], we have chosen the maximum of the optimizers in such cases. In [7], the authors proposed another method for finding the optimum bandwidth for classification. Here, we have also used that algorithm, and the resulting classifier is referred to as the optimal bandwidth classifier.

Our proposed method performed quite well on these data sets. In the case of BUPA liver disorder data, it significantly outperformed its all competitors, whereas in all other cases, it performance was close to the best ones. On the biomedical data set, all methods except the MISE-bandwidth classifier and the adaptive regression method had similar misclassification rates. Error rates of these two classifiers were marginally higher. On the glass data, the MISE-bandwidth classifier had the lowest error rate, but the proposed method could match the performance of this classifier. The bootstrap method, the regression method and the optimal bandwidth classifier had comparatively higher misclassification rates in this example. On the Pima Indian diabetes data set, the optimal bandwidth classifier led to the lowest error rate, but the performance of the proposed method, and that of the regression and bootstrap techniques were also quite competitive, and they significantly outperformed all other kernel density estimate-based classifiers considered here.

B. Classification Involving More Than Two Populations

In this section, we consider six multiclass problems for further illustration of the proposed method. For this purpose, we used both the pairwise and the combined approaches as discussed in Section II. For the analysis of glass data, we considered the same five variables we used before, but here observations from all six classes were taken into consideration. Although there are originally 946 observations in the vehicle data set, we could use only 846, which are available at UCI machine learning repository. In the image data set, although there are 19 measurement variables, we carried out our analysis using only nine variables that were used in [9]. Note that classification problems can be associated with one-response variable regression problems when $J = 2$. For $J > 2$, one can adopt either the pairwise approach or the one verses rest method. In the former case, the results of all pairwise classifications can be combined either using the voting method or using the coupling algorithm [16]. In the later case, however, we do not need voting or coupling. If the ith class $(i = 1, 2, \ldots, J)$ is compared with the rest, and if we define $Y = 1$ when the observation belongs to the ith class and $Y = -1$ otherwise, $\hat{Y}$ gives the estimate of $2p(i|\mathbf{x}) - 1$, from which one can get $\hat{p}(i|\mathbf{x})$. We used both these methods but the results were almost the same. Therefore, here, we report the results only for the pairwise approach, where voting is used to combine the results of all pairwise classifications.

Table III clearly shows the superiority of the proposed methods. On the image segmentation data, both combined and pairwise approaches significantly outperformed all other bandwidth selection methods considered here. On the vowel data, apart from the MISE bandwidth classifier and the square root law, all other methods had similar error rates. All competing methods except adaptive regression had similar error rates on vowel-2 data as well. On the diabetes data and the vehicle data, the regression method and the proposed methods performed significantly better than their competitors. On the glass data, bootstrap method yielded the best performance, but the errors rates of the proposed methods were also quite competitive. To summarize

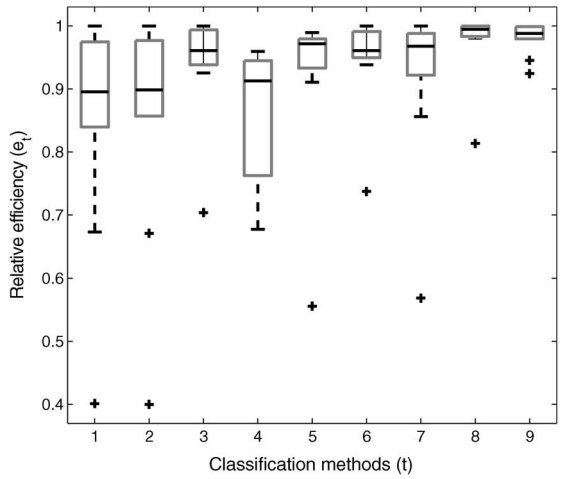

Fig. 1. Relative efficiencies of different methods: 1—MISE bandwidth. 2—Square root law. 3—Regression. 4—Adaptive regression. 5—Cross validation. 6—Bootstrap. 7—Optimal bandwidth. 8—Proposed (pairwise). 9—Proposed (combined).

the overall performance of different methods, we computed their relative efficiencies for different data sets. In a particular example, the relative efficiency of any particular classification method t is defined by the misclassification probability ratio $e_t = \Delta_0/\Delta_t$, where Δ_t is the misclassification rate of the tth classifier and $\Delta_0 = \min_t \Delta_t$. Clearly, in an example, the best classifier will have $e_t = 1$, while other classifiers will have e_t in (0, 1) interval. Smaller values of e_t indicates the lack of efficiency of the classifier t. In each of these ten examples, we computed these relative efficiencies for different methods, and the summarized results are graphically presented by box plots [32] in Fig. 1, which shows the utility of case-specific smoothing techniques.

IV. More on Case-Specific Smoothing

In the previous sections, for classification of a data point $\mathbf{x}$, we have used a single common bandwidth $h_{\mathbf{x}}$ in all directions. If the data cloud is spherical, it seems somewhat reasonable to use the same bandwidth in all directions. This is why standardization is used to make the data cloud spherical in some sense. However, one should notice that, even if the data cloud is spherical, all directions may not always be equally important for classification. For instance, consider a 2-D problem, where the true class boundary $\mathcal{B}$ is of the form $\mathcal{B} = \{(x_1, x_2); x_1 = k\}$. In this set up, x_1 will certainly be more important than x_2 for estimation of Y near the true class boundary. When a small change in x_1 near the class boundary is expected to yield significant change in $E(Y|\mathbf{x})$, $E(Y|\mathbf{x})$ is expected to remain almost constant in x_2. Since Nadaraya–Watson estimate is locally constant, instead of spherical kernel (kernel with the same bandwidth in all directions), in such cases, it will be more helpful to use elliptic kernel function (kernel with different bandwidths in different directions), with elliptic probability contours having more spread along the X_2-axis (i.e., parallel to the class boundary). Naturally, on the one side of the class boundary, we are expected to have the majority of the observations with $Y = 1$, and on the other side the majority of the observations with $Y = -1$. This elliptic kernel function is expected to put higher weight on those data points which are likely to have the same value of Y, and this leads to a better decision rule.

Following this idea, given an observation $\mathbf{x}$, one can look at the rate of change in the expected value of Y near $\mathbf{x}$ along different directions, and use smaller bandwidths along the directions having higher rate of change. Since $E(Y|\mathbf{x})$ is the difference between the posterior probabilities of two classes, higher rate of change gives the indication

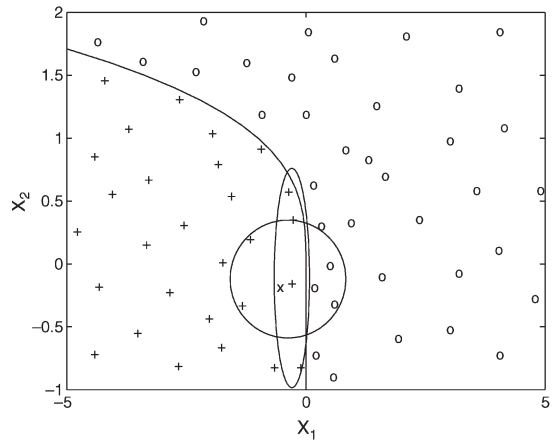

Fig. 2. Spherical and elliptic contours around the observation $\mathbf{x}$.

that we will cross the class boundary relatively quicker if we move along that direction. Naturally, for classification of $\mathbf{x}$, one would like to use a kernel function which puts more weight on the observations lying on the same side of the class boundary, and that is possible if we use elliptic kernels whose probability contours have more spread in the direction parallel to the class boundary (see Fig. 2). If B and W are locally computed between group and within group sum of square matrices near $\mathbf{x}$, eigenvalues and eigenvectors of $M = W^{-1/2}(W + B)W^{-1/2}$ give us some idea about the class separability (local linear separation) near $\mathbf{x}$ along different directions. If we use the bandwidth matrix of the form $H = h^2 M^{-1}$ for some $h > 0$, the kernel function $|H|^{-1/2}K(H^{-1/2}(\mathbf{x} - \mathbf{x}_i))$ will have elliptic probability contours with their spread in different directions (in directions of eigenvectors of M) being proportional to the inverse of the corresponding eigenvalues. Therefore, it shrinks the bandwidth in the direction where the class centroids differ most (i.e., the direction that contains maximum information about class separability). This type of choice for the bandwidth of the kernel function has the same spirit as that of DANN metric considered in [15]. In fact, $\sqrt{(\mathbf{x}_1 - \mathbf{x}_2)' M (\mathbf{x}_1 - \mathbf{x}_2)}$ is the distance between $\mathbf{x}_1$ and $\mathbf{x}_2$ in DANN metric, and the corresponding DANN classifier can be viewed as the proposed kernel classifier for uniform kernel function K and suitable adaptive choice of h^2. For computing the local estimates of B and W near $\mathbf{x}$, we consider only first k_0 neighbors (in terms of Euclidean distance) of $\mathbf{x}$ and assign different weights to them according to their distances. An observation $\mathbf{x}_i$ has weight proportional to $K((\mathbf{x}_i - \mathbf{x})/(\lambda/3))$, where $\lambda = \|\mathbf{x}, \mathbf{x}^{(k_0)}\|$. The choice of the factor 1/3 is mainly motivated by the use of the Gaussian weight function (kernel function), and it allows only the first k_0 neighbors of $\mathbf{x}$ to have significant contributions to B and W. For our data analysis, we chose $k_0 = \max\{n/5, 50\}$, when n was bigger than 50; otherwise, we used $k_0 = \min\{n, 40\}$. Alongwith M, h has to be specified as well. Since different shapes of neighborhoods are used here for different observations, the use of a fixed h does not make sense. Therefore, we went for the spatially adaptive choice of h following the method discussed in Section II.

The importance of using different bandwidths in different directions is quite evident from Table IV, which shows the error rates for spherical and elliptic kernel functions. On the biomedical data, the two-class glass data, the diabetes data and the vehicle data, elliptic kernel led to significantly better performance. It also improved the error rate on the image data and the vowel data. In most of the other cases, error rate remains almost the same. Only in the case of six class glass data, the use of elliptic kernel led to somewhat higher error rates. However, one should note that in the glass data, there are some classes with very few observations (70, 76, 29, 17, 13, and 9 observations in

TABLE IV
AVERAGE MISCLASSIFICATION (IN PERCENT) RATES AND THEIR STANDARD ERRORS FOR SPHERICAL AND ELLIPTIC KERNELS

	Biomed	Glass (2C)	BUPA	Pima	Image	Vowel	Vowel-2	Diabetes	Vehicle	Glass
Spherical (pair.)	16.60 (.13)	20.84 (.24)	32.30 (.14)	26.60 (.07)	5.90	44.59	22.52	10.70 (.18)	20.35 (.07)	31.54 (.10)
Elliptic (pair.)	13.78 (.12)	19.10 (.23)	32.37 (.14)	26.62 (.07)	5.19	41.77	22.22	8.26 (.17)	18.62 (.08)	32.52 (.16)
Spherical (comb.)	16.60 (.13)	20.84 (.24)	32.30 (.14)	26.60 (.07)	6.24	45.24	19.82	10.54 (.17)	20.38 (.07)	31.60 (.10)
Elliptic (comb.)	13.78 (.12)	19.10 (.23)	32.37 (.14)	26.62 (.07)	5.76	38.74	19.82	7.81 (.17)	17.84 (.06)	32.73 (.17)

TABLE V
AVERAGE CPU TIMES (IN SECONDS) REQUIRED FOR DIFFERENT CLASSIFICATION METHODS

	Biomed	Glass(2C)	BUPA	Pima	Image	Vowel	Vowel-2	Diabetes	Vehicle	Glass
MISE-band.	0.28	0.28	1.18	5.14	0.67	1.89	0.62	0.22	3.48	0.15
Sq.root law	0.29	0.28	1.20	5.22	0.86	1.96	0.65	0.22	3.60	0.15
Cross-valid.	0.29	0.30	1.23	5.51	1.94	10.39	3.01	0.32	8.51	0.32
Bootstrap	5.06	5.35	21.44	96.60	28.83	184.38	56.25	5.51	142.23	5.35
Opt-band.	2.17	2.48	6.19	15.88	18.12	75.97	39.99	3.29	29.16	6.96
Regression	0.41	0.45	1.83	9.47	6.49	36.08	7.82	0.66	25.60	0.94
Proposed(p)	0.45	0.27	1.68	15.11	822.42	907.64	44.09	1.02	308.85	6.61
Proposed(c)	0.45	0.27	1.68	15.11	56.88	35.28	5.00	0.26	74.79	0.68

six different classes). After dividing the observations into training and test sets, we had even smaller number of training sample observations from these classes, which made it difficult to get reasonable estimates for B and W. This could be the reason for the poor performance of the elliptic kernel function in this data set. In this correspondence paper, although we have reported the error rates only for the classifiers based on kernel density estimates, for some of these benchmark data sets, performance of other parametric and nonparametric classifiers have been reported in [9] and [23]. Performance of our proposed method is quite comparable to those reported results.

A. Computational Issues

For classification of an observation $\mathbf{x}$, since the proposed method requires the optimization of $\alpha_h(\mathbf{x})$ or $\alpha_h^*(\mathbf{x})$ over several h, it is expected to take more time than a classifier that uses a common h estimated from the training data. However, one should also note that for finding this common h, standard algorithms like cross validation or bootstrap may require significant amount of computations, particularly in the presence of large training data. The proposed method does not need such resampling techniques to predefine the value of h. This helps to reduce the computational cost substantially in such cases.

In Table V, we have reported the average CPU times required by different methods for classification of different data sets. For adaptive regression, these computing times were much higher, and that is why we do not consider them for comparison. For instance, in the case of vowel-2 data set, it took 2544 seconds, while all other methods took less than 60 s. For the proposed method, we report the CPU time for classification based on spherical kernel function. The use of the elliptic kernel led to marginal increase (less than 10% in all cases) in the average CPU time. From Table V, it is quite transparent that computationally it is always advantageous to use the combined approach. The pairwise approach required relatively higher computing time compared to other methods, particularly when we have a bigger test set consisting of observations from several competing populations. However, the combined approach reduced this computing cost substantially, and its average CPU time was comparable to that of the other methods. Only in the case of image data, when we had to classify 2100 test cases based on a training sample of size 210, this computing time was somewhat higher, but it all other cases, it took less time than some of the classifiers that use a common h estimated from the training data. However, as we have mentioned before, a fixed bandwidth classifier requires more time for selecting the value of h, but after that, it classifies the future observations fairly quickly. Therefore, for these classifiers, the online complexity is very low. In these ten data sets, after the selection of h, for classifying all test set observations by a fixed bandwidth classifier, the average computing times were found to be 0.006, 0.004,0.016, 0.081, 0.235, 0.188, 0.062, 0.003, 0.125, and 0.006 s, respectively. On the contrary, the proposed method takes more time (reported in Table V) for classifying the future observations, but it needs no offline calculation to predefine the value of h.

V. CONCLUDING REMARKS

Here, we present a simple and effective algorithm for case-specific smoothing in KDA. Construction of a good classifier needs proper estimation of the class boundary. If the true class boundary is smooth in one part and wiggly in other part, ideally one should use different levels of smoothing in those two parts. Although this kind of adaptive smoothing is quite popular in density estimation and regression problems, the main focus of those adaptive bandwidth selection algorithms is on the accuracy of the corresponding function estimates. However, in practice, it is possible to have poor estimates of density functions (or regression functions) which lead to a good classifier [6], [7]. This is because of the special kind of bias variance decomposition of these function estimates in classification problems and their effect on the misclassification rate as discussed in [6]. Therefore, instead of using the adaptive bandwidth selection methods available for regression and density estimation, one needs to develop a different algorithm for classification problems. Using the simple concept of p-value, here, we present one such algorithm and amply demonstrate its usefulness analyzing several benchmark data sets.

In this correspondence paper, we have assumed normality of the distribution of kernel density estimates in order to find out the p-value and the case-specific choice of h. Since kernel density estimates are

averages of i.i.d. random variables, this is a valid assumption even for moderately large training samples. However, in small sample cases, this assumption may not hold, and one needs to adopt the bootstrap method [5] for this purpose. Although bootstrap is computationally expensive, but in the case of small training set, this computing cost will not be significant. The interval $[h_{\mathrm{L}}(\mathbf{x}), h_{\mathrm{U}}(\mathbf{x})]$ that we have used for maximization of $\alpha_h(\mathbf{x})$ or $\alpha_h^*(\mathbf{x})$, is mainly motivated by Theorem 1, and our empirical experience also supported it. However, this may not be the optimum choice for some of these data sets, where a more judicious choice may further improve the performance of the proposed classifier.

REFERENCES

[1] I. Abramson, "On bandwidth variation in kernel estimates—A square root law," *Ann. Stat.*, vol. 10, no. 4, pp. 1217–1223, Dec. 1982.

[2] T. Bailey and A. Jain, "A note on distance-weighted k-nearest neighbor rules," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, no. 4, pp. 311–313, Apr. 1978.

[3] O. Chapelle, A. Zien, and B. Scholkopf, Eds., *Semi Supervised Learning.* Cambridge, MA: MIT Press, 2006.

[4] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[5] B. Efron, "Estimating the error rate of a prediction rule: Improvement on cross-validation," *J. Amer. Stat. Assoc.*, vol. 78, no. 382, pp. 316–331, Jun. 1983.

[6] J. H. Friedman, "On bias, variance, 0-1 loss, and the curse of dimensionality," *Data Mining Knowl. Discov.*, vol. 1, no. 1, pp. 55–77, 1997.

[7] A. K. Ghosh and P. Chaudhuri, "Optimal smoothing in kernel discriminant analysis," *Stat. Sin.*, vol. 14, no. 2, pp. 457–483, Apr. 2004.

[8] A. K. Ghosh, "On nearest neighbor classification using adaptive choice of k," *J. Comput. Graph. Stat.*, vol. 16, no. 2, pp. 482–502, Jun. 2007.

[9] A. K. Ghosh and S. Bose, "Feature extraction and classification using statistical networks," *Int. J. Pattern Recog. Artif. Intell.*, vol. 21, pp. 1103–1126, 2007.

[10] A. K. Ghosh, "Kernel discriminant analysis using case-specific smoothing parameters," Dept. Math. Stat., IIT Kanpur, India, 2008. Tech. Rep. [Online]. Available: http://www.geocities.com/ghosh_anilk

[11] P. Hall and M. P. Wand, "On nonparametric discrimination using density differences," *Biometrika*, vol. 68, pp. 287–294, 1988.

[12] P. Hall and K.-H. Kang, "Bandwidth choice for nonparametric classification," *Ann. Stat.*, vol. 33, no. 1, pp. 284–306, 2005.

[13] D. J. Hand, *Kernel Discriminant Analysis*. Chichester, U.K.: Wiley, 1982.

[14] W. Hardle and J. S. Marron, "Optimal bandwidth selection in nonparametric regression function estimation," *Ann. Stat.*, vol. 13, no. 4, pp. 1465–1481, Dec. 1985.

[15] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–616, Jun. 1996.

[16] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *Ann. Stat.*, vol. 26, no. 2, pp. 451–471, 1998.

[17] M. C. Jones, J. S. Marron, and S. J. Sheather, "A brief survey of bandwidth selection for density estimation," *J. Amer. Stat. Assoc.*, vol. 91, no. 433, pp. 401–407, Mar. 1996.

[18] D. O. Loftsgaarden and C. P. Quesenberry, "A nonparametric estimate of a multivariate density function," *Ann. Math. Stat.*, vol. 36, no. 3, pp. 1049–1051, Jun. 1965.

[19] H. G. Muller, "Smooth optimum kernel estimators of densities, regression curves and modes," *Ann. Stat.*, vol. 12, no. 2, pp. 766–774, Jun. 1984.

[20] H. G. Muller and U. Stadtmuller, "Variable bandwidth kernel estimators of regression curves," *Ann. Stat.*, vol. 15, no. 1, pp. 182–201, Mar. 1987.

[21] G. E. Peterson and H. L. Barney, "Control methods used in a study of the vowels," *J. Acoust. Soc. Amer.*, vol. 24, no. 2, pp. 175–184, Mar. 1952.

[22] J. Rice, "Bandwidth choice for nonparametric regression," *Ann. Stat.*, vol. 12, no. 4, pp. 1215–1230, Dec. 1984.

[23] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[24] S. R. Sain and D. W. Scott, "On locally adaptive density estimation," *J. Amer. Stat. Assoc.*, vol. 91, no. 436, pp. 1525–1534, Dec. 1996.

[25] W. Schucany, "Adaptive bandwidth choice for kernel regression," *J. Amer. Stat. Assoc.*, vol. 90, no. 430, pp. 535–540, Jun. 1995.

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

[27] S. J. Sheather and M. C. Jones, "A reliable data-based bandwidth selection method for kernel density estimation," *J. R. Stat. Soc., B*, vol. 53, no. 3, pp. 683–690, 1991.

[28] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.

[29] J. Sklansky and G. N. Wassel, *Pattern Classifiers and Trainable Machines*. New York: Springer-Verlag, 1981.

[30] J. G. Stainwalis, "Local bandwidth selection for kernel estimates," *J. Amer. Stat. Assoc.*, vol. 84, no. 405, pp. 284–288, 1989.

[31] G. Terrel and D. W. Scott, "Variable kernel density estimation," *Ann. Stat.*, vol. 20, no. 3, pp. 1236–1265, 1992.

[32] J. W. Tukey, *Exploratory Data Analysis*. New York: Addison-Wesley, 1977.

[33] P. Vieu, "Nonparametric regression: Optimal local bandwidth choice," *J. R. Stat. Soc., B*, vol. 53, no. 2, pp. 453–464, 1991.

[34] M. P. Wand and M. C. Jones, *Kernel Smoothing*. London, U.K.: Chapman & Hall, 1995.

[35] G. N. Wassel and J. Sklansky, "Training a one dimensional classifier to minimize the probability of error," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-2, no. 4, pp. 533–541, Sep. 1972.

[36] Z. Zhang, J. T. Kwok, and D.-Y. Yeung, "Model based transductive learning of the kernel matrix," *Mach. Learn.*, vol. 63, no. 1, pp. 69–101, Apr. 2006.