Multiscale Classification Using Nearest Neighbor Density Estimates

Anil K. Ghosh, Probal Chaudhuri, and C. A. Murthy

***Abstract*—Density estimates based on k-nearest neighbors have useful applications in nonparametric discriminant analysis. In classification problems, optimal values of k are usually estimated by minimizing the cross-validated misclassification rates. However, these cross-validation techniques allow only one value of k for each population density estimate, while in a classification problem, the optimum value of k for a class may also depend on its competing population densities. Further, it is computationally difficult to minimize the cross-validated error rate when there are several competing populations. Moreover, in addition to depending on the entire training data set, a good choice of k should also depend on the specific observation to be classified. Therefore, instead of using a single value of k for each population density estimate, it is more useful in practice to consider the results for multiple values of k to arrive at the final decision. This paper presents one such approach along with a graphical device, which gives more information about classification results for various choices of k and the related statistical uncertainties present there. The utility of this proposed methodology has been illustrated using some benchmark data sets.**

***Index Terms*—Bootstrap, cross validation, misclassification rate, multiscale analysis, posterior probability, p-value, weighted averaging.**

I. Introduction

In CLASSIFICATION problems, one aims to achieve the maximum accuracy in predicting class labels of multivariate observations $\mathbf{x}$. If π_j's are the prior probabilities, and f_j's $(j = 1, 2, \ldots, J)$ are the density functions of J competing classes, Bayes rule is given by $d(\mathbf{x}) = \arg\max \pi_j f_j(\mathbf{x})$. However, in practice, the density functions f_j's are usually unknown, and they are estimated using the training-sample observations. Nearest neighbor density estimation (e.g., see [12] and [21]) is one popular nonparametric method for finding these estimates of population densities. For estimating f_j at a point $\mathbf{x}$, it assumes f_j to be constant over a closed ball (neighborhood) around $\mathbf{x}$. The distance between $\mathbf{x}$ and its k_jth nearest neighbor in the training sample coming from the jth class is taken as the radius of this ball. Consequently, $f_j(\mathbf{x})$ is estimated by $\hat{f}_{j,k_j}(\mathbf{x}) = k_j/n_j V_{j,k_j}(\mathbf{x})$, where n_j is the corresponding training-sample size, and $V_{j,k_j}(\mathbf{x})$ is the volume of the neighborhood. The parameter k_j controls the size of the neighborhood, and consequently the smoothness of the density estimate. As k_j gets larger, the density estimate tends to be "more flat," and hence "more smooth" in some sense. In this paper, we will refer to it as the neighborhood parameter.

The performance of the nearest neighbor density estimates and that of the corresponding classification rule depends on the values of these neighborhood parameters. Existing asymptotic results (e.g., see [12] and [21]) suggest that k_j should vary with n_j in such a way that for every $j = 1, 2, \ldots, J$, k_j should tend to infinity and k_j/n_j should tend to zero as $n_j \to \infty$. However, for moderately large and small sample sizes, there is no theoretical guideline for choosing the optimum value of k_j. In classification problems, since the optimum value of the neighborhood parameter of a class depends also on the competing class densities, it is meaningful to minimize the cross-validated error rate (e.g., see [26] and [32]) $\Delta(k_1, k_2, \ldots, k_J)$ simultaneously with respect to $k_1, k_2, \ldots, k_J$ to find the optimal neighborhood parameters. However, it is computationally difficult to implement this cross-validation method when $J > 2$. It should also be noted that the optimum choice of k_j's is case specific, and it depends on the observation to be classified in addition to depending on the entire training data set. Further, for a specific observation, one may also like to assess the strength of evidence for different classes for varying choices of k_j. Therefore, in classification, instead of relying on a single value of k_j, it may be of more use to look at the results for different scales of smoothing to come up with the final decision.

This paper presents a multiscale approach, where classification results for multiple values of neighborhood parameters are studied simultaneously in order to build up a more informative classification procedure. These results are presented in a two-dimensional plot, which is specific to an observation to be classified. This plot enables an effective visual comparison between the strength of different classes at some particular point of the sample space. Recently, Ghosh *et al.* (see [13]) developed such a visual device for kernel discriminant analysis based on varying choices of bandwidth parameters. Similar multiscale-type techniques were used in [3], [4], and [16] to identify significant features in function estimation problems. The performance of different classification rules obtained for varying choices of neighborhood parameters is judged on the basis of the corresponding estimated misclassification probabilities. These misclassification probabilities can also be presented in two-dimensional plots. All these plots give some useful information for classification, which is combined together to arrive at the final result.

One should also notice that nearest neighbor density estimates depend on the distance function as well. Euclidean metric

Manuscript received February 3, 2005; revised August 3, 2005 and November 28, 2005. This paper was recommended by Associate Editor M. Huber.

A. K. Ghosh is with the Centre for Mathematics and Its Applications, Mathematical Sciences Institute, The Australian National University, Canberra, ACT 0200, Australia (e-mail: anilkghosh@rediffmail.com).

P. Chaudhuri is with the Theoretical Statistics and Mathematics Unit, Indian Statistical Institute, Kolkata 700 108, India (e-mail: probal@isical.ac.in).

C. A. Murthy is with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India (e-mail: murthy@isical.ac.in).

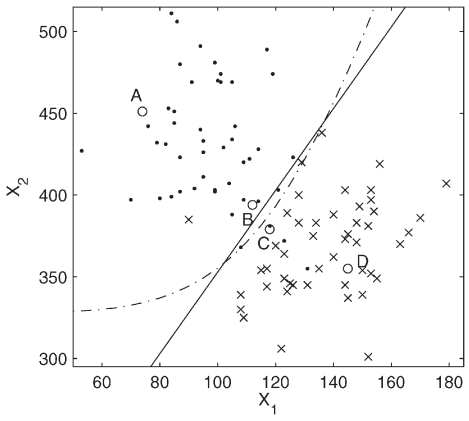


Fig. 1. Scatter plot of salmon data. Class boundaries estimated by LDA1 and QDA.

is the most popular choice for this distance function. One can standardize the data set using an estimate of the class dispersion matrix and compute the density estimate for the standardized variable. The density estimate at the original data point can be obtained from that using a simple transformation formula, where the measurement vector undergoes a linear transformation. Note that the usual nearest neighbor classification rule (e.g., see [5] and [9]) considers the densities of different classes to be constant over a common neighborhood around $\mathbf{x}$, but k-nearest neighbor (k-NN) density estimates allow us to have different shapes of neighborhoods for different populations.

II. Illustrative Example

Let us consider the following example with a salmon data set taken from [20]. This data set consists of 100 bivariate observations on growth ring diameters (freshwater and marine water) of salmon fish taken from Alaskan and Canadian water. A scatter plot of this data set is given in Fig. 1, where dots ("$\cdot$") and crosses ("$\times$") represent the observations coming from Alaskan and Canadian populations, respectively. We chose four out of these 100 observations from different parts of the data (marked by "$\circ$" in the figure) for which the class information is known and classified them using the remaining 96 observations. Observations "A" and "B" belong to the Alaskan population, whereas "C" and "D" belong to the Canadian population. Notice that for observations "A" and "D," the evidence in favor of the true class is much stronger than that in the other two cases. As discussed in [20], in this data set, the data distributions for both classes appear to satisfy the assumption of normality, and hence, the traditional methods of linear and quadratic discriminant analysis (LDA and QDA), especially QDA, perform well. In this example, both LDA and QDA could correctly classify all the four observations. The estimated class boundaries for these two methods are given in Fig. 1. In this figure, we observe that "B" and "C" are very close to the class boundary, but they lie on the opposite sides of the separating line (curve). "A" and "D" also belong to opposite sides, but they are far away from the line (curve) of discrimination. So, one should normally expect to have different behaviors of the classification methodology for these four observations.

Using a simple Euclidean distance and a leave-one-out cross-validation technique for minimization of $\Delta(k_1, k_2)$ on this data set, we obtained $k_1 = 3$ and $k_2 = 9$ as the best choice for neighborhood parameters. However, this choice of (k_1, k_2) failed to properly exhibit the difference in the strength of classification. It could correctly classify only three of these four observations. Posterior probability estimates in favor of the Alaskan population were found to be 0.9577, 0.7910, 0.7222, and 0.1132, respectively, for "A," "B," "C," and "D." This cross-validation method could not estimate the class boundary properly, and it classified "B" and "C" to the same class with almost equal posterior estimates. Later in this section and in Section III, we will see that in this case, more improved results can be obtained if we carry out our analysis using multiple values of (k_1, k_2). Since different values of neighborhood parameters correspond to different scales of smoothing, this study using multiple values of (k_1, k_2) will be referred to as multiscale analysis. In multiscale analysis, we measure the strength of evidence for two competing classes for different choices of neighborhood parameters, and they are displayed in a two-dimensional plot. This plot provides an effective visual comparison between the strengths of different classes.

A. Multiscale Analysis Using Posterior Probability

If k_1 and k_2 are used as the neighborhood parameters for the two classes, the estimated posterior probability for the first population is given by $P_{k_1,k_2}(1|\mathbf{x}) = \pi_1 \hat{f}_{1,k_1}(\mathbf{x})/\{\pi_1 \hat{f}_{1,k_1}(\mathbf{x}) + \pi_2 \hat{f}_{2,k_2}(\mathbf{x})\}$, where π_1 and π_2 are the prior probabilities of the two classes. Varying the values of k_1 and k_2, we get a sequence of posterior estimates for each observation. The plots in the first row of Fig. 2 show the grayscale values of these posterior estimates, where white color denotes the highest posterior (i.e., $P_{k_1,k_2}(1|\mathbf{x}) = 1$) and black color denotes the lowest posterior (i.e., $P_{k_1,k_2}(1|\mathbf{x}) = 0$) in favor of population 1 (i.e., Alaskan population in this example). The intensity of the color varies with the magnitude of the posterior estimate. As expected, for observation "A," we observe very light color over the entire plot from which the decision in favor of the Alaskan population is quite transparent. The same is true for observation "D," where the plot shows a strong evidence in favor of the Canadian population. However, for the other two cases, the decisions are not that clear. In these cases, we observe gray color over the entire plot with a little lighter or darker shade in various regions, which gives a clear indication of borderline cases. One may also notice the dominance of lighter shades in case of observation "B" and that of darker shades for "C." This gives us some useful idea about the final classification of these observations.

B. Multiscale Analysis Using a p-Value-Type Measure

Instead of posterior probabilities, one may plot the probability function $\Psi_{k_1,k_2}(1|\mathbf{x}) = P\{\pi_1 \hat{f}_{1,k_1}(\mathbf{x}) > \pi_2 \hat{f}_{1,k_2}(\mathbf{x})\}$ as well. This probability function Ψ can be viewed as a one-sided p-value for testing the hypothesis $H_0 : E\{\pi_1 \hat{f}_{1,k_1}(\mathbf{x})\} \leq E\{\pi_2 \hat{f}_{1,k_2}(\mathbf{x})\}$ against $H_a : E\{\pi_1 \hat{f}_{1,k_1}(\mathbf{x})\} > E\{\pi_2 \hat{f}_{1,k_2}(\mathbf{x})\}$, and that is why we chose to call it a p-value-type measure (see [13] for a discussion on a similar p-value in the context of

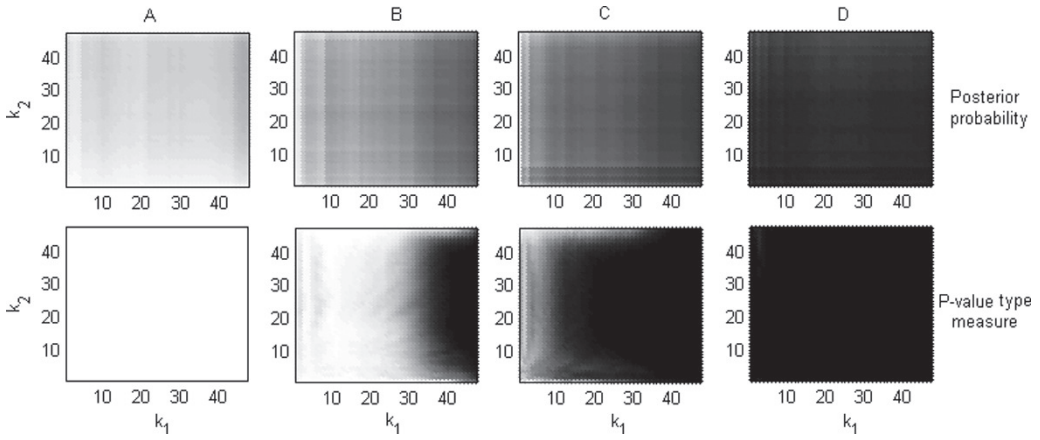


Fig. 2. Estimated posterior probabilities and p-value-type measures at four selected data points.

discriminant analysis using kernel density estimates). Clearly, high and low values of $\Psi_{k_1,k_2}(1|\mathbf{x})$ give decisions in favor of the first and the second populations, respectively. We know that if k_j varies with n_j in such a way that $k_j \to \infty$ and $k_j/n_j \to 0$ as $n_j \to \infty$, for any given $\mathbf{x}$, $\hat{f}_{j,k_j}(\mathbf{x})$ converges to the true density function $f_j(\mathbf{x})$ in probability if f_j is continuous (e.g., see [12] and [21]). If this condition holds for both $j = 1$ and $j = 2$, the posterior-probability estimate $P_{k_1,k_2}(1|\mathbf{x})$ being a continuous function of $\hat{f}_{1,k_1}(\mathbf{x})$ and $\hat{f}_{2,k_2}(\mathbf{x})$, converges (in probability) to the true posterior as $\min\{n_1, n_2\} \to \infty$ (e.g., see [29]). When $\mathbf{x}$ lies on the common support of f_1 and f_2, this true posterior is a value in the range (0, 1). However, one should also notice that under the above condition, $\pi_1\hat{f}_{1,k_1}(\mathbf{x}) - \pi_2\hat{f}_{2,k_2}(\mathbf{x})$ converges to $\theta(\mathbf{x}) = \pi_1 f_1(\mathbf{x}) - \pi_2 f_2(\mathbf{x})$ in probability. Therefore, for any given $\mathbf{x}$, depending on whether $\theta(\mathbf{x})$ is positive or negative, $\Psi_{k_1,k_2}(1|\mathbf{x})$ converges (in probability) either to 1 or to 0. Thus, the use of this p-value-type measure is expected to give more weight in favor of the winning class, and thereby, it would sharpen the plot by enhancing the black and white contrast.

However, it is difficult to get any closed-form expression for this probability function Ψ. Here, we estimate it using bootstrap samples (e.g., see [8]) from the two populations. Given a training sample of n_1 observations from population 1, there are n_1 distances between the training-set observations and the specific data point to be classified. From this set of n_1 numbers (distances), we choose a sample of size n_1 using with replacement technique, and treat them as the distances of the specific data point from the newly generated training set. We sort these new n_1 distances to find the order statistics, and density estimates for different values of k_1 can be easily computed from this sorted sequence. Similarly, the density estimate for the competing population can be computed for different values of k_2 as well. We generate these random samples for a large number of times and the proportion of times, where $\pi_1\hat{f}_{1,k_1}(\mathbf{x})$ is larger than $\pi_2\hat{f}_{1,k_2}(\mathbf{x})$, is taken as the bootstrap estimate for $\Psi_{k_1,k_2}(1|\mathbf{x}) = P\{\pi_1\hat{f}_{1,k_1}(\mathbf{x}) > \pi_2\hat{f}_{1,k_2}(\mathbf{x})\}$.

As expected, the resulting plot sharpens the picture, and thereby makes it more effective for visualization. The plots in the second row in Fig. 2 show the grayscale values of estimated $\Psi_{k_1,k_2}(1|\mathbf{x})$ for different values of k_1 and k_2, when 1000 bootstrap samples are used for estimation. Once again, decisions for observations "A" and "D" become quite clear from these plots, as we observe white or black color over the entire region. For the other two observations, which lie near the class boundary, we observed white as well as black shades in the plots, which give indications about borderline cases. Percentages of white or black regions also give some indications about the final classification of "B" and "C," and these indications are clearer than those obtained in corresponding posterior probability plots.

In the case of bivariate data, we can get an idea about the location of a data point from the scatter plot itself. However, in a high-dimensional problem, it is difficult to visualize whether a data point is near the class boundary or if it is far away from it. The plots of posterior probability and p-value are helpful in such situations. Using these plots, one can easily compare the strength of the competing classes at a given data point and form an idea about whether it is a borderline case or a clear-cut one.

III. Aggregation of Results

To make the final decision for an observation, one should also consider the statistical uncertainties associated with classification results. One should rely more on those neighborhood parameters, which lead to lower misclassification probabilities. Here, we estimate these misclassification probabilities by the leave-one-out cross-validation method, and the grayscale values for the corresponding probabilities of correct classification are presented in a two-dimensional plot (see Fig. 3). For better visualization, we rescale these probabilities to have minimum value 0 and maximum value 1. Clearly, light and dark color point towards low misclassification probability (high probability for correct classification) and high misclassification probability, respectively. Thus, the plot shows the preferable values of (k_1, k_2) for a given data set. The values of (k_1, k_2) lying in the light colored region are the values one should rely more, and the user should assign more weight on them while aggregating the classification results for different (k_1, k_2) to arrive at the final decision.

A natural way to aggregate the results for different classifiers is to take the weighted average of the estimated posterior probabilities. Well-known aggregation techniques like bagging (e.g., see [1]), boosting (e.g., see [11] and [27]), and arcing

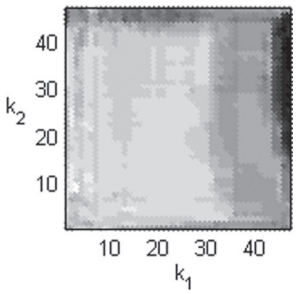


Fig. 3. Probability of correct classification (rescaled).

(e.g., see [2]) adopt similar ideas for aggregating the results of several classifiers. Recently, Holmes and Adams [18] used a logistic regression setup for aggregating the results of several nearest neighbor classifiers based on the Markov–Chain Monte Carlo technique (e.g., see [15]). In another paper (see [19]), they considered similar regression setup and used an iterative reweighted least square method (e.g., see [22]) for aggregation. Ghosh *et al.* [14] also proposed an aggregation technique for combining the results of usual nearest neighbor classifiers. However, none of these authors deal with nearest neighbor density estimates, which are used in this paper.

If $w(k_1, k_2)$ is used as the weight for the pair of neighborhood parameters (k_1, k_2), the weighted posterior for the first population can be expressed as

$$P^{w}_{k_1,k_2}(1|\mathbf{x}) = \frac{\sum_{k_1=1}^{n_1-1}\sum_{k_2=1}^{n_2-1} w(k_1,k_2) P_{k_1,k_2}(1|\mathbf{x})}{\sum_{k_1=1}^{n_1-1}\sum_{k_2=1}^{n_2-1} w(k_1,k_2)}.$$

One would expect the weight $w(k_1, k_2)$ to be higher for those values of (k_1, k_2), which lead to lower misclassification probability $\Delta(k_1, k_2)$. Unlike what happens in bagging (e.g., see [1]), which uses equal weights for all classifiers to be aggregated, we propose to use a weight function that would decrease gradually as the misclassification rate increases. Boosting (e.g., see [11] and [27]) uses a similar idea for aggregation, where the weight $w = \log_e\{(1-\Delta)/\Delta\}$ is used. This logarithmic weight function used in boosting decreases with misclassification probability at a very slow rate, and it is our empirical experience that for aggregation of classifiers based on nearest neighbor density estimates, this choice of the weight function sometimes fails to appropriately down weight the poor classifiers leading to unsatisfactory classification results. Instead, if one uses a Gaussian-type weight function, the poor classifiers would be down weighted appropriately. Here, we estimate $\Delta(k_1, k_2)$ by the leave-one-out cross-validation technique and define the weight function as

$$w(k_1,k_2) = \begin{cases} e^{-\frac{1}{2}\frac{\left(\widehat{\Delta}(k_1,k_2)-\Delta_0\right)^2}{\frac{\Delta_0(1-\Delta_0)}{N}}}, & \text{if } \frac{\left(\widehat{\Delta}(k_1,k_2)-\Delta_0\right)^2}{\frac{\Delta_0(1-\Delta_0)}{N}} \le \tau \text{ and} \\ & \widehat{\Delta}(k_1,k_2) < \min\{\pi_1,\pi_2\} \\ 0, & \text{otherwise} \end{cases}$$

where N is the training-sample size and $\Delta_0 = \min_{k_1,k_2} \widehat{\Delta}(k_1, k_2)$ (see also [13]). Notice that Δ_0 and $\Delta_0(1-\Delta_0)/N$ can be viewed as estimates for the mean and the variance of the empirical misclassification rate of the best classifier based on nearest neighbor density estimates when such a classifier is used to classify N independent observations. The constant τ determines the maximum amount of deviation from Δ_0 in a standardized scale beyond which the weighting scheme ignores the classifiers by putting zero weight on them. Clearly, $\tau = 0$ corresponds to the situation of putting all the weights only on those classifiers for which $\widehat{\Delta}(k_1, k_2) = \Delta_0$, and when this optimal classifier is unique, the resulting aggregation method becomes equivalent to the usual leave-one-out cross-validation technique. However, in the context of multiscale analysis, it is more meaningful to consider some positive value of τ so that the results of other classifiers can be taken into consideration. However, because of the choice of the Gaussian weight function above, one does not need to consider a value of τ larger than three in practice. Moreover, the pair (k_1, k_2) is allowed to have positive weight only if the performance of the corresponding classifier is better than that of a trivial classifier (i.e., the misclassification rate is smaller than both the prior probabilities). Of course, the above choice of weight function is somewhat subjective, and one may use many other suitable functions as well. Our empirical experience suggests that the final result is not very sensitive to the weighting procedure as long as any reasonable weight function, which decreases appropriately at an exponential or high-order polynomial rate as the estimated error rate increases, is used. A comparative empirical study of bagging, boosting, and other ensemble methods is available in [24].

In the example with salmon data, our aggregation method led to posterior estimates 0.8804, 0.6447, 0.3994, and 0.1046, respectively, for "A," "B," "C," and "D." Note that unlike the cross-validated choice of (k_1, k_2), this aggregated classifier correctly classified all the four observations, and it could properly exhibit the difference in the strengths of classification as well. Using this method, we could arrive at different classification results for "B" and "C," which one would normally expect from the scatter plot in Fig. 1.

As we have mentioned before, in the presence of $J(J > 2)$ competing populations, it becomes computationally difficult to evaluate the misclassification rates $\Delta(k_1, k_2, \ldots, k_J)$ for a whole range of different values of $k_1, k_2, \ldots, k_J$. In such cases, we adopt a pairwise approach, which splits a multiclass problem into several two-class problems taking a pair of classes at a time, and thereby makes it computationally tractable. It has been stated earlier that the optimal neighborhood parameter of a class density estimate not only depends on the population itself but also on its competing class densities. Therefore, it is more useful to consider different neighborhood parameters for a class density estimate when it is compared with different competing class densities. Our pairwise approach allows this flexibility, and at the same time, it makes it possible to present the results of multiscale analysis in two-dimensional plots even when there are more than two competing populations. After all pairwise comparisons are carried out, the results are combined by the method of majority voting (e.g., see [10]) to come up

with the final decision. Instead of voting, one may use the method of pairwise coupling (e.g., see [17]) as well, but the latter one is computationally costlier.

IV. Results From the Analysis of Benchmark Data Sets

In this section, we use some benchmark data sets to compare the performance of our proposed aggregation method. Along with the error rates of our method, we also report the misclassification rates of other classifiers based on nearest neighbor density estimates that use a single value of k_j for each population. These methods require the optimum value of k_j to be estimated. One can estimate this value by optimizing some suitable criterion based on marginal density estimates. Least square cross-validation LSCV (e.g., see [28] and [31]) is one such technique, where we look for minimization of the mean integrated square error (MISE $= \int\{\hat{f}_{j,k_j}(\mathbf{x}) - f_j(\mathbf{x})\}^2 d\mathbf{x}$) of the density estimates. In practice, a cross-validated unbiased estimate of MISE is used for this minimization. However, since this method involves the calculation of $\int \hat{f}^2_{j,k_j}(\mathbf{x})d\mathbf{x}$, it is computationally difficult to use it for high-dimensional problems. Instead, one may select the optimal bandwidth using a likelihood cross-validation (LCV) (e.g., see [31]) technique. LCV selects the optimum k_j by maximizing the log-likelihood score $\sum_{i=1}^{n_j} \log\{\hat{f}^{(-i)}_{j,k_j}(\mathbf{x}_{ji})\}$, where $\mathbf{x}_{ji}$ is the ith observation from the jth class, and $\hat{f}^{(-i)}_{j,k_j}(\mathbf{x}_{ji})$ is the nearest neighbor density estimate of $f_j(\mathbf{x}_{ji})$ obtained by the leave-one-out method. One should notice that both these cross-validation methods select the optimum neighborhood parameters based on marginal population distributions only, whereas in a classification problem, the optimum value of k_j not only depends on the jth population but may also depend on its competing population densities. Therefore, in practice, it is more useful to minimize the cross-validated misclassification rate $\Delta(k_1, k_2, \ldots, k_J)$ simultaneously with respect to $k_1, k_2, \ldots, k_J$. To differentiate this method from other cross-validation techniques (LSCV and LCV), we will refer to it as $\mathrm{CV}_{\mathrm{class}}$. As we have mentioned before, due to computational burden, it is very difficult to minimize $\Delta(k_1, k_2, \ldots, k_J)$ when $J > 2$. In such cases, we apply $\mathrm{CV}_{\mathrm{class}}$ on each pair of classes, and the results are then combined by the method of majority voting (e.g., see [10]). However, no such voting method is required for LSCV and LCV.

In all cases, we first standardized the data sets using estimated dispersion matrices, and then used the Euclidean metric to find the nearest neighbor density estimate for the standardized variable. The density estimate at the original data point was obtained from that using a simple transformation formula. To compute $f_j(\mathbf{x})$ $(j = 1, 2, \ldots, J)$ at a new data point $\mathbf{x}$, at first, it is premultiplied by a matrix S_j for standardization. The density estimate at $S_j\mathbf{x}$ is computed using the standardized observations of the jth class, and then it is multiplied by the determinant of S_j to get the density estimate at the original data point $\mathbf{x}$. One can either use an estimate of the pooled dispersion matrix $\mathbf{\Sigma}$ for standardization of all data points (i.e., $S_j = \widehat{\mathbf{\Sigma}}^{-1/2}$ for all $j = 1, 2, \ldots, J$) or the user may use the estimates of class dispersion matrices $\mathbf{\Sigma}_j$ for standardization of observations in different classes (i.e., $S_j = \widehat{\mathbf{\Sigma}}_j^{-1/2}$ for $j = 1, 2, \ldots, J$). However, one may notice that in the former case, multiplication by the determinant of $\widehat{\mathbf{\Sigma}}$ is not necessary for the classification purpose. Whether one should use the pooled dispersion matrix or separate dispersion matrices for different classes depends on the data set to be classified as well as on the classification method to be used. Here, we used both types of standardization for LCV, $\mathrm{CV}_{\mathrm{class}}$, and our proposed method, and in each case, we reported the result for that one, which led to a lower misclassification rate. For finding the error rates of our aggregation method, we have always used $\tau = 3$. This choice of τ is mainly motivated from the use of the Gaussian-type weight function. Throughout this section, training-sample proportions of different classes were used as their prior probabilities.

Instead of fixing the values of individual k_j's as it is done in the case of LCV, LSCV, or $\mathrm{CV}_{\mathrm{class}}$, if we fix the value of $k = \sum k_j$ and use the same neighborhood for all populations, it leads to the usual k-NN classification rule (e.g., see [5] and [9]). This k-NN classifier is very popular among statisticians as well as in machine-learning communities, and it has been extensively investigated in the literature (e.g., see [6], [7], [33], and [34]) from various perspectives. To facilitate the comparison with our aggregated classifier, in this paper, we report the error rates of k-NN classification based on Mahalanobis distance (which is equivalent to Euclidean distance after standardization) and the leave-one-out cross-validated choice of k.

We have used 15 data sets in this section. Among them, the salmon data have been described earlier in Section II. A description of the vowel data-1 (we have used two different data sets on the vowel recognition problem and denoted them as vowel data-1 and vowel data-2) is given in [25]. Phoneme data and its description are available at http://www.dice.ucl.ac.be/neural-nets/Research/ Projects/ELENA/elena.htm. The rest of the data sets and their descriptions are available either at the University of California—Irvine (UCI) machine-learning repository (http://www.ics.uci.edu) (see [23]) or at the Carnegie Mellon University (CMU) data archive (http://lib.stat.cmu.edu). Six of these data sets (synthetic data, vowel data-1, satimage data, vowel data-2, letter-recognition data, and sonar data) have specific training and test samples. In all other cases, we divided the whole data set randomly into two parts to form the training and the test sets. The sizes of the training and the test sets in each partition are given in Table I. This random partitioning was carried out 1000 times to generate 1000 different training and test samples. Average test-set error rates over these 1000 partitions are reported for different methods along with their corresponding standard errors. Only in the case of phoneme data were 100 random partitions used. For synthetic data, vowel data-1, satellite image data (satimage data), vowel data-2, letter-recognition data, and sonar data, which have specific training and test sets, we have reported the test-set misclassification errors for different classifiers. For these six data sets, if a classifier leads to a test-set error rate p, the corresponding standard error is taken as $\sqrt{p(1-p)/N_{\mathrm{t}}}$, where N_{t} is the size of the test sample. All these results are given in Table I below. Due to computational

TABLE I
AVERAGE TEST-SET MISCLASSIFICATION RATES (IN PERCENTAGE) AND THEIR STANDARD ERRORS FOR DIFFERENT CLASSIFICATION METHODS

Data sets	No. of features	No. of classes	Sample size		Nearest neighbor	LCV	CV_{class}	Weighted averaging
			Training	Test				
Salmon	2	2	50	50	8.98 (0.10)	10.69 (0.10)	9.04 (0.10)	7.91 (0.10) $^{\bullet 3}$
Synthetic^{+}	2	2	250	1000	11.70 (1.02)	13.70 (1.09)	11.00 (0.99)	10.30 (0.96) $^{\bullet 1}$
Vowel-1^{+}	2	10	338	333	17.72 (2.09) $^{\bullet 1}$	23.72 (2.33)	20.72 (2.22)	18.62 (2.13)
Biomed	4	2	100	94	17.95 (0.11)	14.82 (0.09) $^{\bullet 2}$	14.95 (0.09)	17.81 (0.10)
Iris	4	3	100	50	2.58 (0.07)	3.92 (0.07)	2.75 (0.07)	2.28 (0.06) $^{\bullet 3}$
Satimage$^{\circ,+}$	4	6	4435	2000	15.35 (0.81)	19.80 (0.89)	15.45 (0.81)	15.25 (0.81) $^{\bullet 1}$
Phoneme	5	2	3000	2404	12.08 (0.06)	12.07 (0.06)	12.14 (0.06)	11.80 (0.06) $^{\bullet 3}$
Diabetes	5	3	100	45	10.01 (0.13)	10.58 (0.12)	8.92 (0.12)	8.53 (0.12) $^{\bullet 3}$
Crab	5	4	100	100	6.60 (0.07)	7.48 (0.07)	6.54 (0.07)	5.63 (0.06) $^{\bullet 3}$
Pima Indian	8	2	300	468	25.97 (0.05)	31.04 (0.06)	25.56 (0.05)	25.04 (0.04) $^{\bullet 3}$
Vowel-2^{+}	10	11	528	462	46.75 (2.32) $^{\bullet}$	46.75 (2.32) $^{\bullet}$	47.19 (2.32)	46.75 (2.32) $^{\bullet}$
Wine	13	3	100	78	2.19 (0.05)	2.29 (0.05)	2.20 (0.05)	1.84 (0.04) $^{\bullet 3}$
Aust. Credit	14	2	300	390	13.88 (0.04) $^{\bullet 3}$	27.37 (0.34)	14.24 (0.04)	14.19 (0.04)
Letter Recog.$^{+}$	16	26	16000	4000	4.43 (0.33)	4.25 (0.32)	4.15(0.32) $^{\bullet}$	4.25 (0.32)
Sonar^{+}	20	2	104	104	17.31 (3.71)	10.58 (3.02) $^{\bullet}$	15.38 (3.54)	16.34 (3.63)

$^{+}$ Data sets have specific training and test sets.

$^{\circ}$ Four central pixel values were used for classification.

$^{\bullet}$ Best error rate but this error rate is not significantly lower than that of the any competing classifier.

$^{\bullet t}$ Best error rate and this error rate is significantly lower than that of other t competing classifiers ($t = 1, 2, 3$).

difficulties, we could use LSCV only for two-dimensional problems. On the synthetic data, this method achieved the same misclassification rate as obtained by LCV. On the salmon data and the vowel data-1, it led to error rates of 8.49% and 30.0%, respectively, with corresponding standard errors of 0.11% and 2.51% in the respective cases.

In ten out of 15 data sets, our proposed aggregation method led to the lowest misclassification rates among the classifiers considered here. Moreover, using the corresponding standard errors reported inside the braces, in most of these cases, its error rate was found to be statistically significantly lower than that of the other classifiers. On as many as seven data sets, it significantly outperformed all the three competing classifiers. Apart from biomedical data, in all other cases, if not significantly better, the aggregation method could achieve comparable error rates to that of the CV_{class} technique. Only in the case of Australian data and vowel data-1 could the popular nearest neighbor classifier perform better than our proposed method, though in the latter case, the difference was statistically insignificant. The performance of LCV was not satisfactory at all. In ten out of 15 data sets, it led to the highest error rates among the competing classifiers.

If we take a closer look at the biomedical data, we can notice some observations, which are sparsely distributed, and they are almost outliers with respect to their true classes. Fig. 4 shows the results of multiscale analysis for one such observation. Recall that the biomedical data set does not have specific training and test sets, and we formed those sets by randomly partitioning the data. In each partition, training sets were formed by taking 35 observations from class-1 and 65 observations from class-2, while the rest of the observations were used as test cases for that partition. Fig. 4 shows the results of multiscale analysis for a test case originally taken from class-1. The first two plots on the left (rescaled version of probability of correct classification and that of weight function) give an idea about the range of values of k_1 and k_2 one should look at. A white spot on the bottom left corner also indicates that the cross-validation method selected $k_1 = k_2 = 1$ as the best value of neighborhood parameters. Light colors on the bottom-left corners in the plots of posterior probabilities and p-values suggest that the cross-validated choice of neighborhood parameter yields the correct result in this case. However, if one looks at these two plots more carefully, the user can notice that apart from some small values of k_1 and k_2, in all other cases, there is a strong evidence in favor of the other class. From these plots, it is quite clear that in the training sample, there are only two or three data points from the true class corresponding to that test case, which are in the vicinity of the test case, while the rest of the observations from the true class are far away from it. This gives an indication that the test case was really an outlier from class-1, which is its correct class. Our aggregation method failed to classify this outlier correctly. Because of the presence of some outliers like this in the data set, our aggregation technique led to poor performance in this data set.

Breiman [1] argued that there is not much gain in combining the classifiers using subsampling techniques like bagging or boosting when the classifiers are stable. In the context of usual nearest neighbor classification, Shalak [30] suggested to combine the classifiers only when they have reasonable amount of diversity among themselves. Since the classifier based on

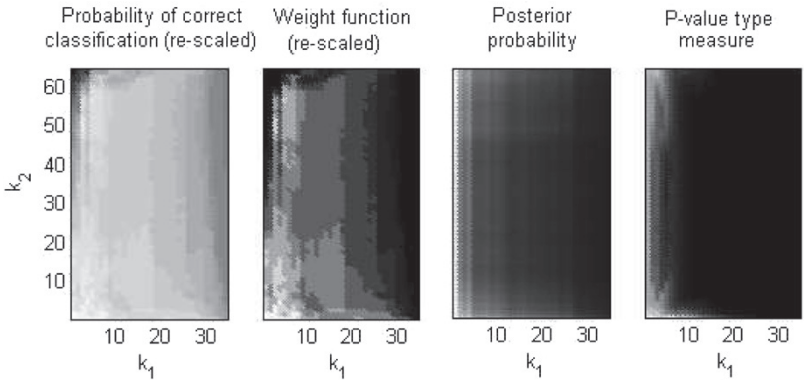


Fig. 4. Multiscale analysis of an observation from biomedical data.

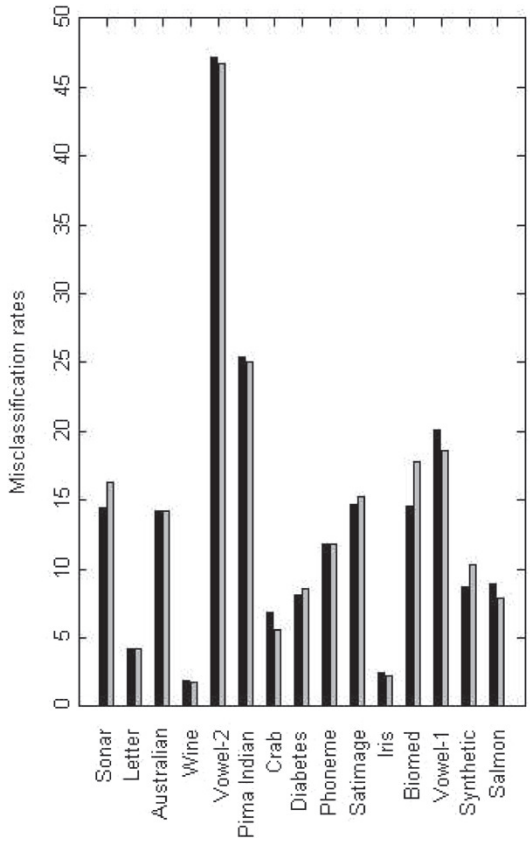
TABLE II
CPU TIMES (IN SECONDS) REQUIRED FOR DIFFERENT CLASSIFICATION METHODS

Data sets	No. of features	No. of classes	Sample size		Nearest neighbor	LCV	CV_{class}	Weighted averaging	
			Training	Test				original	truncated
Salmon	2	2	50	50	0.003	0.002	0.004	0.007	0.003
Synthetic^{+}	2	2	250	1000	0.172	0.141	0.406	0.796	0.187
Vowel-1^{+}	2	10	338	333	0.140	0.047	0.534	1.224	0.235
Biomed	4	2	100	94	0.012	0.013	0.026	0.032	0.017
Iris	4	3	100	50	0.009	0.008	0.028	0.038	0.018
Satimage$^{\circ,+}$	4	6	4435	2000	22.015	11.438	442.927	966.422	31.711
Phoneme	5	2	3000	2404	11.937	13.922	321.344	349.828	18.125
Diabetes	5	3	100	45	0.009	0.009	0.030	0.035	0.017
Crab	5	4	100	100	0.014	0.013	0.037	0.053	0.026
Pima Indian	8	2	300	468	0.157	0.324	0.704	0.900	0.404
Vowel-2^{+}	10	11	528	462	0.437	0.625	3.220	3.814	1.543
Wine	13	3	100	78	0.019	0.047	0.088	0.096	0.075
Aust. Credit	14	2	300	390	0.170	0.680	1.149	1.386	0.866
Letter Recog.$^{+}$	16	26	16000	4000	424.063	429.062	10098.719	11206.604	1789.494
Sonar^{+}	20	2	104	104	0.031	0.141	0.187	0.203	0.184

$^{+}$ Data sets have specific training and test sets.

$^{\circ}$ Four central pixel values were used for classification.

nearest neighbor density estimates with nearly the same values of k_j's are expected to produce similar classification results, it seems that there is not much gain in combining them when all weights are distributed among only few classifiers having almost similar values of k_1 and k_2. In such cases, the aggregation method is expected to perform as good as the leave-one-out cross-validation technique. However, it should be noted that in terms of misclassification rates, one would normally expect to gain by combining the classifiers, and the diversity among different classification rules can be viewed as a measure of the extent to which the error rates can be improved. For widely different values of k_1 and k_2, the classifiers based on nearest neighbor density estimates are expected to have reasonable diversity among themselves. While the classifiers with smaller values of k_1 and k_2 are expected to catch the local patterns of the measurement space, more global patterns will be extracted by the larger values. Therefore, when these classifiers with widely different values of k_j's produce comparable misclassification rates (i.e., they have positive weights in aggregation), the proposed aggregation method, being a combination of widely different classifiers, is expected to perform better. For instance, in the case of salmon data, the light color over a large region in Fig. 3 indicates that in terms of misclassification rate, the classifiers with some smaller values of k_1 and k_2 are as good as the classifiers with some larger values of k_1 and k_2. Therefore, in this case, our proposed aggregation method considered the results of all these different classifiers by putting positive weights on them. As a result, the weighted averaging method led to significant improvement in the misclassification rate of the resulting classifier. Not only on salmon data, the aggregation technique could achieve significantly better performance than the CV_{class} method on six other data sets as well. Only in the case of biomedical data did the presence of several outliers in test sets lead to significantly higher misclassification rate for the aggregation method.



Data sets	Weighted averaging	
	original	truncated
Salmon	7.91(0.10)	8.89(0.09)
Synthetic	10.30(0.96)	8.70(0.89)
Vowel-1	18.62(2.13)	20.12(2.20)
Biomed	17.81(0.10)	14.62(0.09)
Iris	2.28(0.06)	2.53(0.06)
Satimage	15.25(0.81)	14.65(0.79)
Phoneme	11.80(0.06)	11.80(0.06)
Diabetes	8.53(0.12)	8.10(0.12)
Crab	5.63(0.06)	6.30(0.06)
Pima Indian	25.04(0.06)	25.41(0.04)
Vowel-2	46.75(2.32)	47.19(2.32)
Wine	1.84(0.04)	1.92(0.05)
Aust. Credit	14.19(0.04)	14.21(0.04)
Letter Recog.	4.25(0.32)	4.18(0.32)
Sonar	16.31(3.63)	14.42(3.44)

Fig. 5. Misclassification rates for aggregated nearest neighbor classifiers.

A. Computational Complexity and Related Issues

Note that, from a computational perspective, the dimension d is involved only for computing the pairwise distances between the data points, and this is common for all the nearest neighbor methods. Thus, for our complexity calculations, we shall start from the stage when all pairwise distances are given to us. We shall also assume that for all $j = 1, 2, \ldots, J$, n_j/N (where $N = \sum n_j$) remains bounded away from 0 and 1, i.e., $n_1, n_2, \ldots, n_J$ are of the same asymptotic order $O(N)$. Under this condition, classification using nearest neighbor density estimates requires $O(N^3)$ computations to define the weight function, and after that $O(N^2)$ calculations are required for classification of a new observation. One major problem of the multiscale aggregation method is this computational complexity. In order to reduce this computational cost, instead of aggregating the classifiers for all possible values of (k_1, k_2), one may restrict this aggregation to $k_1 \leq \sqrt{n_1}$ and $k_2 \leq \sqrt{n_2}$. This choice is mainly motivated by the theoretical result that if $k_j \to \infty$ and $k_j/n_j \to 0$ as $n_j \to \infty$, the nearest neighbor density estimates converge to the true density functions in probability. This truncation makes the aggregation technique computationally more efficient. It requires $O(N^2)$ calculations to compute the weight function, while the classification of a new observation requires $O(N)$ computations. In Table II, we report the CPU times for different classification algorithms when they are run on a Pentium IV machine. The computational advantage of the truncated aggregation method over the original aggregation procedure is quite transparent from this table, especially in the cases of large data sets.

Fig. 5 gives a comparison between the performance of this truncated aggregation procedure (indicated by black bars) and that of the original aggregated classifier (indicated by gray bars). From this figure, it is quite evident that the truncation method did a reasonably good job in most of the cases.

V. Conclusion

This paper presents a multiscale approach for classification based on nearest neighbor density estimates. Instead of using a single value of the neighborhood parameter for each class, it studies the results for a sequence of neighborhood parameters simultaneously in order to develop a more informative classification procedure. In practice, the use of fixed values of neighborhood parameters may not work well in different parts of the measurement space. In such cases, it is more useful to consider the results for different levels of smoothing. The multiscale technique adds that flexibility to the classification methodology.

The multiscale method has another useful application in terms of visualization. The plots of p-values and posterior probabilities provide effective visual comparisons between the strengths of different competing classes. These plots give useful information about the distribution of different classes in the vicinity of the observation to be classified, which helps us to form an idea about the location of the data point with reference to the separating surfaces. This makes it easier to identify the borderline cases from the clear-cut ones, which is very helpful in high-dimensional problems, where we cannot use a two-dimensional scatter diagram to visualize

the distributional geometry of data clouds. For classification among several populations, when it is computationally difficult to use usual cross-validation techniques to select optimum neighborhood parameters, our pairwise treatment not only reduces the computational cost significantly, but also facilitates the visual representation of multiscale analysis using two-dimensional plots.

The aggregation method used in this paper is simple in nature. In order to study the robustness of this proposed classification procedure with respect to the choice of τ, we need a thorough analytic and empirical investigation, which is well beyond the scope of this paper. However, the results of our empirical studies reported in this paper indicate that only in one data set out of a total of 15 data sets does the aggregation method lead to a statistically significant deterioration of the performance of the classifier when compared with the performance of the classifier corresponding to $\tau = 0$, which has similar performance as the classifier denoted as $\mathrm{CV}_{\mathrm{class}}$. In 12 of the data sets, the aggregation method improved the performance over the classifier corresponding to $\tau = 0$, and in two cases (i.e., the "sonar data" and the "letter-recognition data"), the aggregation method led to the deterioration of misclassification rates but only by an amount that is statistically insignificant. In the case of "biomedical data," there is a significant rise in the misclassification rate when one uses the proposed aggregation method. In Section IV, we have used the visualization device on an observation from this data set that appears to behave like a typical outlier with respect to its true class, and the presence of such observations in the data has caused poor performance of the aggregation method. It is possible that there is some intrinsic lack of robustness in the proposed aggregation method, which might cause higher misclassification rates when there are such outliers in the data that seem to occur only rarely. As compared to the performance of other classification methods like LCV, LSCV, and the usual nearest neighbor classifier, where fixed values of neighborhood parameters are used over the entire measurement space, the proposed aggregation procedure produced significantly better performance on most of the data sets, while its performance on the other data sets was also quite competitive. In view of the above data analysis, it is appropriate to conclude that aggregation of results for multiple levels of smoothing would usually be better than using a single neighborhood parameter, though the reduction in misclassification rate may not always be statistically significant.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and three anonymous referees who carefully read the earlier version of the paper and provided them with several helpful comments.

REFERENCES

[1] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[2] ——, "Arcing classifiers (with discussion)," *Ann. Stat.*, vol. 26, no. 3, pp. 801–849, Jun. 1998.
[3] P. Chaudhuri and J. S. Marron, "SiZer for exploration of structures in curves," *J. Amer. Stat. Assoc.*, vol. 94, no. 447, pp. 807–823, Sep. 1999.
[4] ——, "Scale space view of curve estimation," *Ann. Stat.*, vol. 28, no. 2, pp. 408–428, Apr. 2000.
[5] 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.
[6] B. V. Dasarathy, "Minimal consistent subset (MCS) identification for optimal nearest neighbor decision system design," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 3, pp. 511–517, Mar. 1994.
[7] S. A. Dudani, "The distance weighted k-nearest neighbor rule," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 4, pp. 325–327, Apr. 1976.
[8] B. Efron and R. Tibshirani, *An Introduction to Bootstrap*. New York: Chapman & Hall, 1993.
[9] E. Fix and J. L. Hodges, Jr., "Discriminatory analysis, nonparametric discrimination, consistency properties," USAF School Aviation Med., Randolph Field, TX, Project 21-49-004, Feb. 1951. pp. 261–279.
[10] J. H. Friedman, "Another approach to polychotomous classification," Dept. Stat., Stanford Univ., Stanford, CA, Oct. 1996. Tech. Rep.
[11] J. H. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression : A statistical view of boosting (with discussion)," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, Apr. 2000.
[12] K. Fukunaga and L. D. Hostetler, "Optimization of k-nearest neighbor density estimates," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 3, pp. 320–326, May 1973.
[13] A. K. Ghosh, P. Chaudhuri, and D. Sengupta, "Classification using kernel density estimates: Multi-scale analysis and visualization," *Technometrics*, vol. 48, no. 1, pp. 120–132, Feb. 2006.
[14] A. K. Ghosh, P. Chaudhuri, and C. A. Murthy, "Visualization and aggregation of nearest neighbor classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1592–1602, Oct. 2005.
[15] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. London, U.K.: Chapman & Hall, 1996.
[16] F. Godtliebsen, J. S. Marron, and P. Chaudhuri, "Significance in scale space for bivariate density estimation," *J. Comput. Graph. Stat.*, vol. 11, no. 1, pp. 1–22, Mar. 2002.
[17] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *Ann. Stat.*, vol. 26, no. 2, pp. 451–471, Apr. 1998.
[18] C. C. Holmes and N. M. Adams, "A probabilistic nearest neighbor method for statistical pattern recognition," *J. R. Stat. Soc., Ser. B*, vol. 64, no. 2, pp. 295–306, May 2002.
[19] ——, "Likelihood inference in nearest-neighbor classification methods," *Biometrika*, vol. 90, no. 1, pp. 99–112, Mar. 2003.
[20] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
[21] D. O. Loftsgaarden and C. P. Quesenberry, "A nonparametric estimate of multivariate density function," *Ann. Math. Stat.*, vol. 36, no. 3, pp. 1049–1051, Jun. 1965.
[22] P. McCullagh and J. Nelder, *Generalized Linear Models*. London, U.K.: Chapman & Hall, 1989.
[23] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, (1998), *UCI Repository of Machine Learning Databases*, Irvine: Dept. Inf. Comput. Sci., Univ. California. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[24] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999.
[25] G. E. Peterson and H. L. Barney, "Control methods used in a study of vowels," *J. Acoust. Soc. Amer.*, vol. 24, no. 2, pp. 175–185, Mar. 1952.
[26] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1996.
[27] R. E. Schapire, Y. Fruend, P. Bartlett, and W. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *Ann. Stat.*, vol. 26, no. 5, pp. 1651–1686, Oct. 1998.
[28] D. W. Scott, *Multivariate Density Estimation: Theory, Practice and Visualization*. New York: Wiley, 1992.
[29] R. Serfling, *Approximation Theorems of Mathematical Statistics*. New York: Wiley, 1980.
[30] D. B. Shalak, "Prototype selections for composite nearest neighbor classifiers," Ph.D. dissertation, Dept. Comput. Sci., Univ. Massachusetts, Amherst, 1996.
[31] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.
[32] M. Stone, "Asymptotics for and against cross-validation," *Biometrika*, vol. 64, no. 1, pp. 29–35, Mar. 1977.
[33] R. R. Yager, "Using fuzzy methods to model nearest neighbor rules," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 4, pp. 512–525, Aug. 2002.
[34] T. P. Yunck, "A technique to identify nearest neighbors," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 10, pp. 678–683, Oct. 1976.

Anil K. Ghosh received the B.Sc. (Hons) degree in statistics from the University of Calcutta, India, in 1996 and the M.Stat. and Ph.D. degrees from the Indian Statistical Institute (ISI), Kolkata, India, in 1998 and 2004, respectively.

During 2004–2005, he was a Postdoctoral Research Fellow at the Institute of Statistical Science, Academia Sinica, Taiwan, R.O.C., and a Visiting Scientist at the Theoretical Statistics and Mathematics Unit, ISI. Currently, he is a Research Associate at the Mathematical Sciences Institute, Australian National University, Canberra, Australia. His research interests include pattern recognition, robust statistics, nonparametric smoothing, and machine learning.

Probal Chaudhuri received the B.Stat. (Hons) and M.Stat. degrees in statistics from the Indian Statistical Institute (ISI), Kolkata, India, in 1983 and 1985, respectively, and the Ph.D. degree in statistics from the University of California, Berkeley, in 1988.

He worked as a member of the Faculty in the University of Wisconsin, Madison, for some time before joining the faculty of ISI, in 1990. He is currently a Professor at the Theoretical Statistics and Mathematics Unit of ISI. His research interests include nonparametric and robust statistics, statistical analysis of molecular data, pattern recognition, and image processing.

Dr. Chaudhuri was elected a Fellow of the Indian Academy of Sciences in 2003 and a Fellow of the Institute of Mathematical Statistics (USA) in 2005. He was awarded the Shanti Swarup Bhatnagar Award for Science and Technology by the Government of India in 2005.

C. A. Murthy received the B.Stat. (Hons), M.Stat., and Ph.D. degrees from the Indian Statistical Institute (ISI), Kolkata, India, in 1979, 1980, and 1989, respectively.

He visited the Michigan State University, East Lansing, in 1991–1992 for six months, and the Pennsylvania State University, University Park for 18 months in 1996–1997. He is a Professor and Head of the Machine Intelligence Unit of ISI. His fields of research interest include pattern recognition, image processing, machine learning, neural networks, fractals, genetic algorithms, wavelets, and data mining.

Dr. Murthy received the best paper award in 1996 in Computer Science from the Institute of Engineers, India. He received the Vasvik award along with his two colleagues in Electronic Sciences and Technology in 1999. He is a fellow of the National Academy of Engineering, India.