Multi-scale classification using nearest neighbor density estimates

Anil K. Ghosh[1], Probal Chaudhuri[2] and C. A. Murthy[3]

[1]**Institute of Statistical Science, Academia Sinica,**
128, Academia Road, Sec 2, Taipei-11529, Taiwan.
[2]**Theoretical Statistics and Mathematics Unit,** [3]**Machine Intelligence Unit,**
Indian Statistical Institute, 203, B. T. Road, Calcutta-700108, India.
E-mail : anilkghosh@rediffmail.com, probal@isical.ac.in, murthy@isical.ac.in

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 also depends 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 article 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. Utility of this proposed methodology has been illustrated using some benchmark data sets.

Index terms : cross-validation, misclassification rate, multi-scale analysis, posterior probability, p-value, weight function.

1 Introduction

In classification problems, one tries 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, the Bayes rule is given by $\mathtt{d}(\mathbf{x}) = \arg\max \pi_j f_j(\mathbf{x})$. However, in practice, these density functions f_j are usually unknown, and they are estimated using the training sample observations. Nearest neighbor density estimation [13], [18] 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_j-th nearest neighbor in the training sample coming from the j-th 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 estimates. As k_j gets larger, the density estimate tends to be "more flat" and hence "more smooth" in some sense. In this article, we will refer to it as the neighborhood parameter.

Performance of the nearest neighbor density estimates and that of the corresponding classification rule depend on the values of these neighborhood parameters. Existing asymptotic results [13], [18] 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 cases, 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 on the competing class densities, it is meaningful to minimize the cross-validated error rate [21], [25] $\Delta(k_1, k_2, \ldots, k_J)$ 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 article presents a multi-scale 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 region of the sample space. Recently, Ghosh, Chaudhuri and Sengupta [14] developed such a visual device for kernel discriminant analysis based on varying choices of bandwidth parameters. Similar multi-scale type techniques were used in [3], [4], [15] to extract significant features in a function estimation problem. Performance of different classification rules obtained for varying choices of neighborhood parameters are judged on the basis of the corresponding estimated misclassification probabilities. These misclassification probabilities are also 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 is the most popular choice for this distance function. Of course, one can also standardize the data set using an estimate of the class dispersion matrix and compute the density estimate for the standardized variable. 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 [10], [5] considers the densities of different classes to be constant over a common neighborhood around $\mathbf{x}$, but k-nearest neighbor density estimates allow for different shapes of neighborhoods for different populations.

2 An illustrative example

Let us consider the following example of salmon data set taken from [17]. This data set consists of 100 bivariate observations on growth ring diameters (freshwater and marine water) of salmon fish of Alaskan and Canadian water. A scatter plot of this data set is given in Figure 2.1, where dots ('·') and crosses ('×') 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 '∘' in the figure) for which the class information is known and classified them using the remaining 96 observations. Observation 'A' and 'B' belong to the Alaskan population whereas 'C' and 'D' belong to the other class. One should notice that in cases of observations 'A' and 'D', the evidence in favor of the true class is much stronger than that in the other two cases. As it has been discussed in [17], in this data set the data distributions for both the classes appear to satisfy the assumption of normality, and hence the traditional methods of linear and quadratic discriminant analysis (LDA and QDA), especially QDA, performs 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 Figure 2.1. 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 two 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.

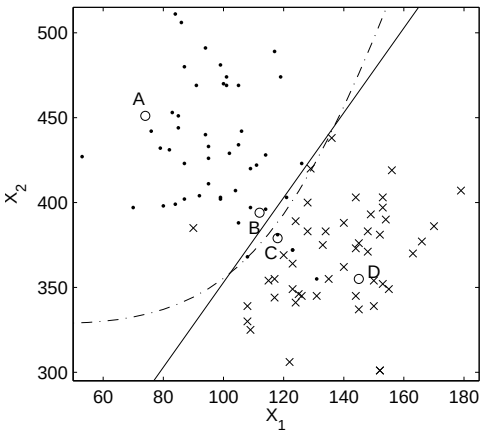

Figure 2.1 : Scatter plot of Salmon data : class boundaries estimated by LDA and QDA.

Using simple Euclidean distance and 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. But 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 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 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 multi-scale analysis. In multi-scale 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.

2.1 Multi-scale analysis using posterior probability

If k_1 and k_2 are used as the neighborhood parameters for the two classes, estimated posterior probability for the first population is given by $P_{k_1,k_2}(1 \mid \mathbf{x}) = \pi_1 \hat{f}_{1,k_1}(\mathbf{x}) / \{\pi_1 \hat{f}_{1,k_1}(\mathbf{x}) + \pi_2 \hat{f}_{2,k_2}(\mathbf{x})\}$, 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 Figure 2.2 show the grey scale values of these posterior estimates, where white color denotes the highest posterior [i.e. $P_{k_1,k_2}(1 \mid \mathbf{x}) = 1$] and black denotes the lowest posterior [i.e. $P_{k_1,k_2}(1 \mid \mathbf{x}) = 0$] in favor of population-1 (i.e., Alaskan population in this example). Intensity of the color varies with the magnitude of the posterior estimates. As expected, for observation 'A', we observe very light color over the entire plot from which the decision becomes quite transparent. The same is true for observation 'D' where the plot shows a strong evidence in favor of Canadian population. However, for the other two cases, decisions are not that clear. In these cases, we observe grey color over the entire plot with a little lighter or darker shade in various regions, which gives clear indication of border line 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 rough idea about the final classification of these observations.

2.2 Multi-scale analysis using a p-value type measure

Instead of posterior probabilities one may plot the probability function $\Psi_{k_1,k_2}(1 \mid \mathbf{x}) = P\{\pi_1 \hat{f}_{1,k_1}(\mathbf{x}) > \pi_2 \hat{f}_{1,k_2}(\mathbf{x})\}$ 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's why we chose to call it as a p-value type measure (see Ghosh, Chaudhuri and Sengupta, 2004 for discussion on a similar p-value in the context of kernel discriminant analysis). Clearly, high and low values of $\Psi_{k_1,k_2}(1 \mid \mathbf{x})$ 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. If this condition holds both for $j = 1$ and $j = 2$, it is easy to see that the posterior probability estimate $P_{k_1,k_2}(1 \mid \mathbf{x})$ converges (in probability) to the true posterior as $min\{n_1, n_2\} \to \infty$. Note that when $\mathbf{x}$ lies on the common support of f_1 and f_2, depending on the location of $\mathbf{x}$, this true posterior is a value in the range (0,1). But under the same condition, it is easy to verify that the corresponding p-value $\Psi_{k_1,k_2}(1 \mid \mathbf{x})$ converges to the 0-1 indicator function $I\{\pi_1 f_1(\mathbf{x}) > \pi_2 f_2(\mathbf{x})\}$ in probability. Therefore, use of this p-value type measure is expected to give more weight on the

winning class and thereby sharpen the plot by enhancing its contrast.

However, it is difficult to get any closed form expression for this probability function Ψ. Here, we estimate it using bootstrap samples (see e.g., [8], [9]) from the two populations. As expected, the resulting plot sharpens the picture and thereby makes it more effective for visualization. The plots in the second row in Figure 2.2 show the grey scale values of $\Psi_{k_1,k_2}(1 \mid \mathbf{x})$ for different values of k_1 and k_2. 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 border line cases. Percentage of white or black regions also suggests about final classifications, and these suggestions are clearer than those obtained in corresponding posterior probability plots.

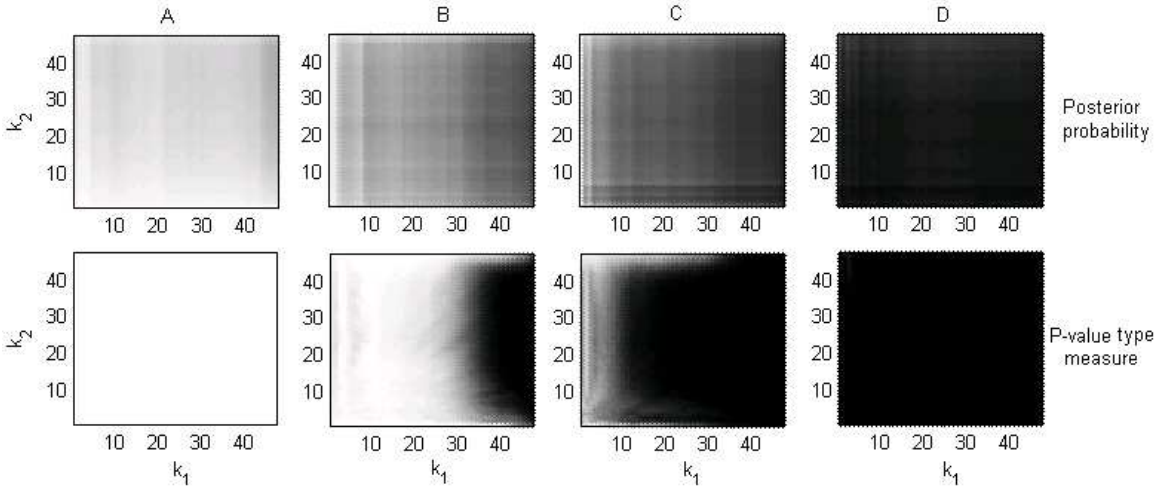

Figure 2.2 : Estimated posterior probabilities and p-value type measures at four selected data points.

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

3 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 leave-one-out cross-validation methods, and the corresponding probabilities of correct classification are presented in a two dimensional plot (see Figure 3.1), where white color points out the regions with low misclassification rates. For better visualization, we re-scale the misclassifica-

tion probabilities to have minimum value 0 and maximum value 1. This plot helps to find out the preferable values of (k_1, k_2), and the estimated posteriors are aggregated accordingly to reach the final decision.

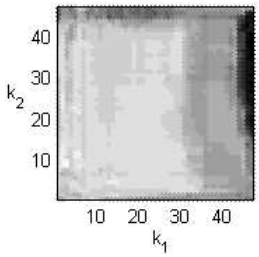

Figure 3.1 : Probability of correct classification (re-scaled).

A natural way to aggregate the results for different classifiers is to take the weighted average of estimated posterior probabilities. The weighted posterior for the first population can be expressed as

$$P^w_{k_1,k_2}(1 \mid \mathbf{x}) = \sum_{k_1} \sum_{k_2} w(k_1, k_2) P_{k_1,k_2}(1 \mid \mathbf{x}) \Big/ \sum_{k_1} \sum_{k_2} w(k_1, k_2),$$

where $w(k_1, k_2)$ is the weight for the pair of neighhood parameters (k_1, k_2). Weights should be higher for those values of (k_1, k_2) which lead to lower misclassification probability $\Delta(k_1, k_2)$. Well known aggregation techniques like bagging [1], boosting [12], [22] and arcing [2] adopt similar idea, where different weights are assigned to different classifiers based on their estimated misclassification rates. In this present article, we use a simple aggregation procedure for combining the results of classifiers based on nearest neighbor density estimates. Here, we estimate $\Delta(k_1, k_2)$ by leave-one-out cross-validation and define the weight function to be

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

where N is the training sample size and $\Delta_0 = \min_{k_1,k_2} \widehat{\Delta}(k_1, k_2)$ (see also [14]). 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 $\widehat{\Delta}(k_1, k_2) = \Delta_0$. But in the context of multi-scale analysis, it is more meaningful to consider some positive value of τ. Because of the choice of above Gaussian weight function, one does not have to consider a value of τ larger than 3 in practice. However, 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 much sensitive to the weighting procedure as long

as any reasonable weight function (which decreases appropriately at an exponential or higher order polynomial rate as the estimated error rate increases) is used.

In the example on salmon data, this 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 should normally expect from the scatter plot in Figure 2.1.

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 multi-class 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. This pairwise approach allows this flexibility, and at the same time it makes it possible to present the results of multi-scale analysis in two-dimensional plots when there are several competing populations. After all pairwise comparisons are carried out, the results are combined by the method of majority voting [11] to come up with the final decision. Instead of voting, one may use the method of pairwise coupling [16] as well.

4 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 criterions based on marginal density estimates. Least square cross validation $LSCV$ [23], [24] is one such technique, where we look for minimization of 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 like to select the optimal bandwidth using likelihood cross-validation (LCV) [24] technique. LCV selects the optimum k_j by maximizing the loglikelihood score $\sum_{i=1}^{n_j} log\{\hat{f}^{(-i)}_{j,k_j}(\mathbf{x}_{ji})\}$, where x_{ji} is the i-th observation from the j-th 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 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 j-th population itself but it 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 w.r.t. $k_1, k_2, \ldots, k_J$. To differentiate this method from other cross-validation techniques ($LSCV$ and LCV), we will refer to it as CV_{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 CV_{class} on each pair of classes and the results are then combined by the method of majority voting [11]. However, no such voting is required for $LSCV$ and LCV.

In all cases, we first standardized the data set using the usual moment based estimate of dispersion matrix and then used Euclidean metric to find out the nearest neighbor density estimate for the standardized variable. Density estimate at the original data point was obtained from that using a simple transformation formula. One can either use an estimate of the pooled dispersion matrix for standardization of all data points or the user may use the estimates of class dispersion matrices for standardization of observations in different classes. Choice of this standardization technique depends on the data set to be classified. Here, we used both types of standardization and reported the better one. For finding the error rates of our aggregation method, we have always used $\tau = 3$. 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 CV_{class}, if we fix the value of $k = \sum k_j$ and use the same neighborhood for all populations, it leads to usual k-nearest neighbor (k-NN) classification rule [10], [5]. This k-NN classifier is very popular among the statisticians as well as in machine learning communities, and it has been extensively investigated in the literature [27], [7], [6], [26] from various perspectives. One of the major issue in k-nearest neighbor classification is to choose the distance metric and the value of k. To facilitate the comparison with our aggregated classifier, in this article we report the error rates of k-NN classification based on Mahalanobis distance [19] (which is equivalent to use Euclidean distance after standardization) and leave-one-out cross-validated choice of k.

We have used 12 data sets in this section. Among them salmon data has been described earlier in Section 2. Description of vowel data-1 (we have used two different data sets on vowel recognition problem and denoted them as vowel data-1 and vowel data-2) is given in [20]. The rest of the data sets and their descriptions are available at either at UCI machine learning repository (http://www.ics.uci.edu) or at CMU data archive (http://lib.stat.cmu.edu). Four of these data sets (synthetic data, vowel data-1, vowel data-2 and sonar data) have separate training and test samples. In all other cases, we divided the whole data randomly into two parts to form the training and the test sets. The sizes of the training and the test set are given in Table 4.1. This random partitioning is 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. For synthetic data, vowel data-1, vowel data-2 and sonar data, which have separate training and test sets, we report the test set misclassification errors for different classifiers. If a classifier leads to a test set error rate p, the corresponding standard error is taken as $\sqrt{p(1-p)/N_t}$,

where N_t is the size of the test sample. All these results are given in Table 4.1 below. Due to computational difficulties, we could use $LSCV$ only for two dimensional problems. On synthetic data, this method achieved the same misclassification rate as obtained by LCV. On salmon data and vowel data-1, it led to error rates of 8.49% and 30.0%, respectively, with the corresponding standard errors of 0.11% and 2.51% in the respective cases.

Data sets	d	J	Sample size		k-NN	LCV	CV_{class}	Weighted averaging
			Training	Test				
Salmon	2	2	50	50	8.98 (0.10)	10.69 (0.10)	9.04 (0.10)	7.93 (0.10)
Synthetic	2	2	250	1000	11.70 (1.02)	13.70 (1.09)	11.00 (0.99)	10.30 (0.96)
Vowel-1	2	10	338	333	17.72 (2.09)	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)	14.95 (0.09)	17.81 (0.10)
Iris	4	3	75	75	4.38 (0.06)	5.88 (0.07)	4.58 (0.07)	3.98 (0.06)
Diabetes	5	3	100	45	10.01 (0.13)	10.58 (0.12)	8.92 (0.12)	8.52 (0.12)
Crab	5	4	100	100	6.60 (0.07)	7.48 (0.07)	6.54 (0.07)	5.62 (0.06)
Pima Indian	8	2	300	468	25.97 (0.05)	31.04 (0.06)	25.56 (0.05)	24.87 (0.04)
Vowel-2	10	11	528	462	46.75 (2.32)	46.75 (2.32)	47.19 (2.32)	46.75 (2.32)
Wine	13	3	100	78	2.19 (0.05)	2.29 (0.05)	2.20 (0.05)	1.84 (0.04)
Australian Credit	14	2	300	390	13.88 (0.04)	27.37 (0.34)	14.24 (0.04)	14.19 (0.04)
Sonar	20	2	104	104	17.31 (3.71)	10.58 (3.02)	15.38 (3.54)	16.34 (3.63)

Table 4.1 : Average test set misclassification rates (in percentage) and their standard errors for different classification methods

The figures in Table 4.1 clearly indicate the superiority of our proposed aggregation method over other cross-validation techniques, where a single value of k_j is used for one class. Apart from sonar and biomedical data, in all other cases, this aggregation method could lead to lower error rates than CV_{class} technique. In view of standard errors reported inside the braces, this reduction in error rates found to be statistically significant for salmon, Iris, crab, wine, chemical and overt diabetes (which is referred to as diabetes data in Table 4.1 and Figure 4.1 later) and Pima Indian data. Only in the case of biomedical data, CV_{class} could produce significantly better performance than our aggregation technique. In all other cases, there was no statistically significant difference between the error rates of these two classifiers.

Overall performance of LCV was poorer than CV_{class}, k-NN and weighted aggregation methods. Only on sonar and biomedical data LCV led to the best performance but in other cases (except vowel data-2 where error rates of all methods were nearly the same) its performance was poorer than that of the other three methods. Apart from these three data seta, in all other cases error rates of weighted aggregation method were lower than that of LCV and in each case this difference was statistically significant as well. On Australian Credit data, LCV performed exceptionally poor.

It is also interesting to note that apart form Australian Credit data and two vowel data sets, our aggregation method could achieve lower misclassification rates than that of the usual k-NN classifier. On salmon, synthetic, Iris, diabetes, crab, Pima Indian, wine and sonar data, these differences between the error rates were statistically significant. On the contrary, only in the case of

Australian Credit data, k-NN classifier performed significantly better than our proposed method. On the rest of the data sets, there was no significant difference between the error rates of these two methods.

4.1 Computational complexity and related issues

Note that, from 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. So, 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 functions, and after that $O(N^2)$ calculations are required for classification of a new observation. 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 converges to the true density function in probability. This truncation makes the aggregation technique computationally more efficient. It requires $O(N^2)$ calculations to compute the weight function, while classification of a new observation requires $O(N)$ computations. Figure 4.1 gives comparison between the performance of this truncated aggregation procedure (indicated by black bars) and that of the original aggregated classifier (indicated by grey bars). From this figure, it is quite evident that the truncation method did a reasonably good job in most of the cases. In almost all the data sets, we observed no statistically significant difference between the error rates of the original aggregation procedure and that of the truncated aggregation method.

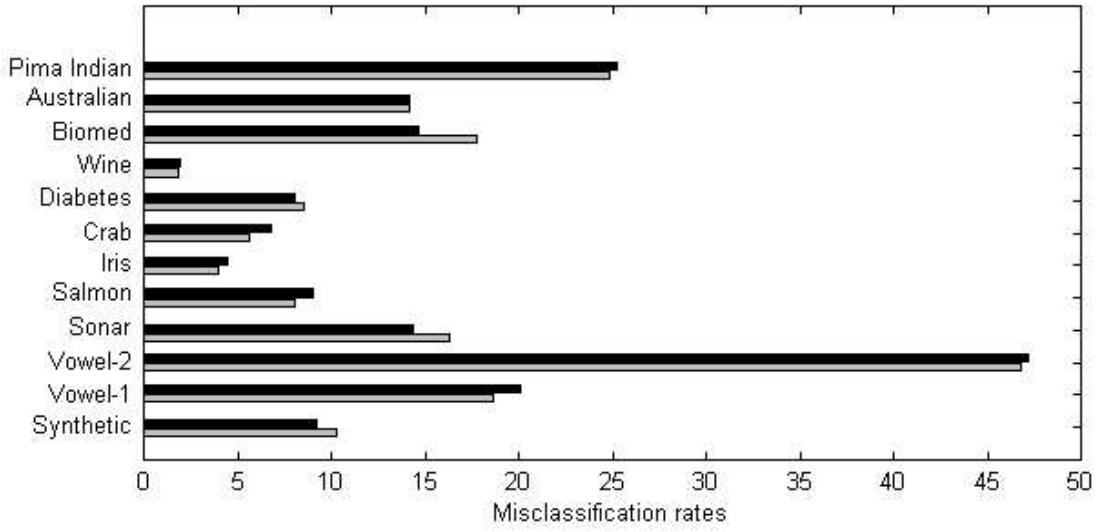

Figure 4.1 : Misclassification rates for aggregated nearest neighbor classifiers.

5 Conclusion

This article presents a multi-scale 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, 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. Multi-scale technique adds that flexibility to the classification methodology.

Multi-scale method has another useful application in terms of visualization. Using the plots of p-values and posterior probabilities, it provides an effective visual comparison 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 ideas about the location of the data point in reference to the separating surface. This makes it easier to identify the border line cases from the clear cut ones, which is very helpful in high dimensional problems, where we can not 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 CV_{class} to select optimum neighborhood parameters, use of pairwise treatment not only reduces the computational cost significantly, but also facilitates the visual representation of multi-scale analysis in a two-dimensional plot.

The aggregation method used in this article is simple in nature. As compared to usual nearest neighbor methods like LCV, $LSCV$, CV_{class} and k-NN, where neighborhood parameters are chosen by cross-validation techniques, this aggregation procedure produced significantly better performance on most of the data sets, while their 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.

References

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

[2] Breiman, L. (1998) Arcing classifiers (with discussion) *Ann. Statist.*, **26**, 801-849.

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

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

[5] Cover, T. M. and Hart, P. E. (1968) Nearest neighbor pattern classification, *IEEE Trans. Info. Theory*, **13**, 21-27.

[6] Dasarathy, B. V. (1994) Minimal consistent subset (MCS) identification for optimal nearest neighbor decision system design. *IEEE SMC*, **24**, 511-517.

[7] Dudani, S. A. (1976) The distance weighted k-nearest neighbor rule. *IEEE SMC*, **6**, 325-327.

[8] Efron, B. (1982) *The Jackknife, the Bootstrap and Other Resampling Plans.* SIAM, Philadelphia.

[9] Efron, B. and Tibshirani, R. (1993) *An Introduction to Bootstrap.* Chapman and Hall, New York.

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

[11] Friedman, J. H. (1996) Another approach to ploychotomous classification. *Tech. Rep., Dept. of Stat., Stanford University.*

[12] Friedman, J. H., Hastie, T. and Tibshirani, R. (2000) Additive logistic regression : a statistical view of boosting (with discussion). *Ann. Statist.*, **28**, 337-407.

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

[14] Ghosh, A. K., Chaudhuri, P. and Sengupta, D. (2004) Classification using kernel density estimates : multi-scale analysis and visualization. *Technometrics* (To appear).

[15] Godtliebsen, F., Marron, J. S. and Chaudhuri, P. (2002) Significance in scale space for bivariate density estimation. *Journal of Computational and Graphical Statistics*, **11**, 1-22.

[16] Hastie, T. and Tibshirani, R. (1998) Classification by pairwise coupling. *Ann. Statist.*, **26**, 451-471.

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

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

[19] Mahalanobis, P. C. (1936) On the generalized distance in statistics. *Proceedings of the National Academy of Sciences, India*, **12**, 49-55.

[20] Peterson, G. E. and Barney, H. L. (1952) Control methods used in a study of vowels. *J. Acoust. Soc. Amer.*, **24**, 175-185.

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

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

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

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

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

[26] Yager, R. R. (2002) Using fuzzy methods to model nearest neighbor rules. *IEEE SMC-B*, **32**, 512-525.

[27] Yunck, T. P. (1976) A technique to identify nearest neighbors. *IEEE SMC*, **6**, 678-683.