ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 37-42

Estimation of Performance Bounds in Supervised Classification

Pierre Comon[1], Jean-Luc Voz[2] and Michel Verleysen[2*]

[1] THOMSON-SINTRA, 525 route des Dolines, BP157,
F-06903 Sophia Antipolis Cedex, France

[2] Université Catholique de Louvain, Laboratoire de Microélectronique - DICE,
3 Place du Levant, B-1348 Louvain-La-Neuve, Belgium

Abstract. The Bayes theory gives the ultimate performances that can be reached in a classification problem. We present in this paper a method that allows to estimate these performance bounds given any finite data set, by building a classifier based on two successive estimations of probability densities, which asymptotically converge to the optimal Bayesian classifier.

1 Introduction

To judge the performances of a classifier, it is necessary to compare them with the ultimate ones given by the Bayes theory. But latter ones are in general unknown because in practice the true data probability density distributions are never known, and only a finite number of training samples (patterns) is available.

This paper presents a method, the "Rough-Refined Estimation" (RRE), that estimates the "best" probability density distributions by proceeding in two stages.

Neural Networks supervised learning techniques generally lead to overfitting if the size of the learning set is small compared to the number of free parameters in the network: even if the Quadratic Error is low, nothing ensures that classification errors are small [3, 7]. On the contrary, the method proposed here provides remarkably good generalization properties versus the size of the learning set.

After a brief summary of the Bayesian theory and kernel density estimators, the trade-off between performance and number of patterns is presented. The RRE algorithm is presented in section 4, and the last section is dedicated to experimental results.

2 Bayesian Framework and Probability Density Kernel Estimators

Assume the problem consists of classifying an observed vector x of $\mathbb{R}^d$ among c classes denoted ω_j. Assume that x is random and that its d components admit

Part of this work has been funded by the ESPRIT-BRA project 6891, ELENA-Nerves II, supported by the Commision of the European Communities (DG XIII).

* Senior Research Assistant of Belgian National Fund for Scientific Research (FNRS).

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 37-42

a joint density $p_x(u)$. If all wrong decisions are given the same penalty, the Bayesian decision will be:

$$Decide\ u \in \omega_s \Leftrightarrow s = Arg \underset{1 \le i \le c}{Max} \{P_i\, p_x(u/\omega_i)\} \tag{1}$$

where $p_x(u/\omega_j)$ is the density of the vector x under the hypothesis that it belongs to class ω_j and P_i is the a priory probability that class ω_i occurs.

It is quite obvious that such an ideal Bayesian solution can be used only if distributions $p_x(u/\omega_j)$, and the c a priori probabilities P_j are known. In the problems we are interested in, it is rarely the case. We rather have at disposal a set of patterns, $A_N = \{x(n), \omega_{(n)}, 1 \le n \le N\}$, where each pattern $x(n)$ belongs to a known class $\omega_{(n)}$. Denote N_j the number of available patterns in class ω_j, $1 \le j \le c$, $\sum_{j=1}^{c} N_j = N$.

One way of performing Bayesian classification is to compute the best estimate of each density $p_x(u/\omega_i)$ in the Mean Square sense with the help of the N_i patterns available. Thus the goal will be to minimize for each class ω_j the error

$$e(N_j) = \int_{\mathbb{R}^d} e(N_j, u/\omega_j)\, du, \tag{2}$$

$$\text{with } e(N_j, u/\omega_j) = E\{[\hat{p}_x(N_j, u/\omega_j) - p_x(u/\omega_j)]^2\}. \tag{3}$$

Because of their nice properties [1], kernel estimators of density have been chosen.

The kernel estimate of a density $p_x(u)$ of a random variable x takes the following general form:

$$\hat{p}_x(N, u) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^d} K\left(\frac{u - x(n)}{h}\right) \tag{4}$$

where $\{x(n), 1 \le n \le N\}$ denote available patterns in a given class. The parameter h is called the *width factor* of the kernel. If h is not allowed to depend on index n, the kernel is referred to as *fixed*, whereas it is referred to as *variable* when the width factor may be different for each $x(n)$. Sufficient and necessary conditions on the series $h(N)$, in order to get the fixed kernel estimator to converge to $p_x(u)$ in the mean square sense may be found in [1, 3].

The kernel is said to be *radial* if K is a function of the norm of its argument only. Better estimates are obtained when the kernel function is not radial, but the computational load is in general much too high, even in moderate dimension. That's why only radial kernels are usually chosen.

3 Large Dimensions and Small Data Sets

An important issue for finite learning sets is to know how many patterns are necessary to reach a given quadratic error, say $e(N) = O(\epsilon)$. One can prove in particular that the minimal error in the fixed kernel case is [3]:

$$\epsilon = O\left(\frac{1}{N}\frac{1}{h^d}\right). \tag{5}$$

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 37-42

Since the optimal value of h is itself of order $N^{-1/(d+4)}$, this yields an order of $N \approx O(\epsilon^{-1-d/4})$ required patterns. The number of patterns required is thus an *exponential* function of the dimension, d. But the coefficient of proportionality cannot be obtained by this approach.

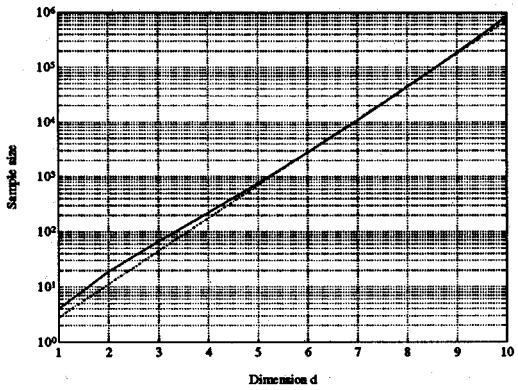

Fig. 1. Sizes of sample required as a function of dimension; solid curve: values given by Silverman; dashdotted; exponential approximation.

Silverman gives, based on extensive experiments [10], the values that are reported in figure 1. This report corresponds to a constant relative error on the density of about 10%, for a fixed Gaussian kernel and a Gaussian underlying density. A simple and sufficiently precise rule can be obtained by the affine approximation:

$$\log_{10} N \approx 0.6\,(d - 0.25). \tag{6}$$

A data set that does not reach the bound (6) will be referred to as *small*, or conversely, the dimension will be said to be *large*.

This value could be thought to be optimistic since the majority of real-world densities to be estimated are non-Gaussian. However, the variable kernel option may considerably diminish the minimal number of patterns. For real-world problems, it can appear that a classifier still performs well even if the number of patterns does not reach (6). In such a case, this simply means that the data set is located in the vicinity of a manifold of dimension smaller than d. A solution that often works if this manifold is linear consists of performing a Principal Component Analysis of the data set; this would reduce the number of attributes while keeping the performances intact.

4 Optimal Kernel, RRE algorithm

Since our concern is mainly to deal with *finite* databases, asymptotic results [6, 4, 5] are not sufficient. Our goal is to design consistent estimators that perform

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 37-42

correctly for limited learning set sizes. In particular, the positivity of the function K will be assumed, whereas it is not required in asymptotic theorems [1, 6]. It will be for instance also required for the kernel function to be concentrated about the origin, and have a variable width factor. The quality of kernel estimators is very sensitive to the width factor when data sets are of moderate size. Previous works have proposed optimal values of h for fixed kernels by approximating the error $e(N)$, resorting to a Taylor expansion of $p_x(u)$ [1, 3, 2].

But since essentially variable kernels will be used in our case, an optimal value that is dependent on u has been calculated, and minimizes $e(N,u)$ for each point u:

$$h(N,u)^{d+4} = \frac{d}{N} \frac{p_x(u)}{\Delta p_x(u)^2} \int_{\mathbb{R}^d} K(w)^2 \, dw, \tag{7}$$

where $\Delta p_x(u)$ denotes the Laplacian of $p_x(u)$. Of course, this expression is not directly applicable, even if it allows in principle to obtain optimal kernel width in the least squares sense, because it involves again the true function $p_x(u)$ which is unknown.

This problem is solved in two steps. First it is resorted to a rough estimator whose width factors are not optimal to compute an estimate of $p_x(u)$ and its Laplacian. Then, this estimates are used to compute the optimal variable width in equation (7).

One simple and efficient solution is to use a k-Nearest Neighbors method to obtain the width factors of the rough kernel estimator : $h(n) = D_k(x(n))$, the distance between vector $x(n)$ and its k^{th} nearest neighbor. This estimate satisfies all the expected properties for the width factor. In practice, k has been empirically chosen as $k = round\left(N^{0.4}\right)$.

Assume radial kernels of the form

$$K(w) = B\, e^{-[Aw^t w]^g} \tag{8}$$

where g is a real number in $(0.5, \infty)$ setting the rate at which the kernel function drops off and coefficients A and B are determined so as to have a unit sum of the density and a unit variance [5]:

$$A = \frac{b}{cd} \; ; \; B = \frac{g\, a\, b^{d/2}}{c^{1+d/2}\,(\pi d)^{d/2}} \tag{9}$$

$$\text{with } a = \Gamma(d/2),\ b = \Gamma((d+2)/2g),\ c = \Gamma(d/2g).$$

The Gaussian kernel ($g = 1$) has been for instance one of our choices.

An estimation of $\Delta p_x(u)$ can then be obtained by calculating the Laplacian of the rough density estimator $\hat{p}_x(N,u)$ via:

$$\Delta\hat{p}_x(N,u) = \frac{1}{N} \sum_{n=1}^{N} \frac{2gA}{h^{d+2}} \left[2g\{Av^tv\}^{2g-1} - (d+2g-2)\,\{Av^tv\}^{g-1}\right] K(v) \tag{10}$$

$$\text{where } v = \tfrac{u-x(n)}{h}.$$

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 37-42

The last term in (7) is called the asymptotic variance coefficient, and takes the following form for kernels (8):

$$\int_{\mathbb{R}^d} K(w)^2 \, dw = 2^{-d/2g} \, B. \tag{11}$$

The refinement procedure may be run more than once. A few simulation examples have shown convergence for less than four runs.

5 Experimental results

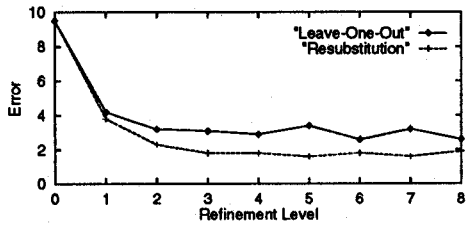

Fig. 2. Errors versus refinement level for the complete database (683 patterns).

A number of simulations have been carried out on a real-world dataset publically available for classification problems [9, 8]. The data are obtained from a clinical study of breast cancer. There are 2 classes (with a 65.5%-34.5% class distribution) and 9 numerical attributes (integer range 1-10) per pattern. The reported studies [8] obtain an error rate of 7.8% to 4.1% with the holdout method on selected testset ranging from 33% to 50% of the patterns.

Figure 2 shows the leave-one-out and resubstitution errors versus the refinement level for the whole database. Since these methods respectively provide an upper and a lower bound of the error probability, the true performance lies in between. This can be verified on figure 3, where the holdhout error averaged for five different partitions is very near from the leave-one-out results averaged on the five learnsets. The generalisation properties of our method are illustrated in table 1 where the average holdout errors, computed after four refinement steps, are reported for decreasing learnset sizes. This shows that variable Gaussian kernel classifiers based on the refinement technique presented here can indeed perform very well even for very small databases.

Table 1. Averaged Holdout error versus the size of the learning set

Size	300	200	100	50	30	20	10
Error	3.15	3.35	3.73	4.12	4.66	4.90	6.03

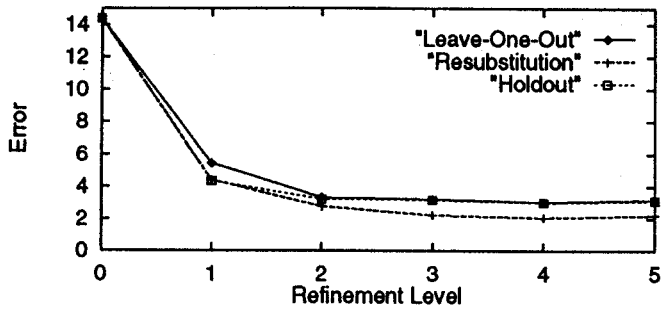

Fig. 3. Errors versus refinement level averaged over five different partitions (learnset: 283 patterns; testset: 400 patterns).

6 Conclusion

The goal of the algorithm presented was to estimate the ultimate classification bounds that can be achieved given a finite data set (containing both learning and test patterns). The excellent generalisation properties also encourage its use to devise classifiers with limited memory capabilities. For the moment, the major inconvenience of the RRE algorithm is its rather heavy computational load.

References

1. T. Cacoullos. Estimation of a multivariate density. *Annals of Inst. Stat. Math.*, 18:178–189, 1966.
2. P. Comon. Classification bayesienne distribuee. *Revue Technique Thomson CSF*, 22(4):543–561, 1990.
3. P. Comon. Classification supervisee par reseaux multicouches. *Traitement du Signal*, 8(6):387–407, December 1991.
4. V.A. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory Proba. Appl*, 14:153–158, 1969.
5. K. Fukunaga and D.M. Hummels. Bayes error estimation using Parzen and k-NN procedures. *IEEE Trans. PAMI*, 9(5):634–643, September 1987.
6. D.J. Hand. *Kernel Discriminant Analysis*. RSP press, 1982.
7. C. Jutten and P. Comon. Neural Bayesian classifier. In Mira Cabestany Prieto, editor, *New Trends in Neural Computation*, number 686, pages 119–124, Berlin, June 1993. Springer-Verlag Lecture Notes in Computer Sciences.
8. O. L. Mangasarian and W.H. Wolberg. Cancer diagnosis via linear programming. In *SIAM News*, volume 23, pages 1–18, September 1990.
9. P. M. Murphy and D.W. Aha. *UCI repository of machine learning databases*. Irvine, University of California, Department of Information and Computer Science (anonymous ftp to ics.uci.edu in pub/machine-learning database).
10. B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.