ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 153-160

Suboptimal Bayesian classification by vector quantization with small clusters

Jean-Luc Voz, Michel Verleysen[*]
Philippe Thissen[†] and Jean-Didier Legat

Université Catholique de Louvain, Microelectronics Laboratory
3 Place du Levant, B-1348 Louvain-La-Neuve, Belgium
tel:+ 32 (10) 472551, fax:+32 (10) 478667,
e-mail:voz@dice.ucl.ac.be

Abstract. Multi-dimensional classification based on the Bayes criterion minimizes the probability of misclassification. In order to apply this criterion, one has to know or to evaluate the probability densities of each class of data. Parzen windows or probabilistic neural networks may be used to estimate these probability densities; however, the number of operations involved in such process is prohibitive for large databases. The proposed algorithm shows how to apply vector quantization techniques to reduce the size of the learning set, while keeping sufficiently accurate estimations of probability densities. The problem of the width of the kernels used in the estimation is addressed by making the hypothesis of small clusters after quantization.

1. Introduction

In multi-dimensional classification tasks, the challenge is to attribute a class label to a vector presented to the system, which previously "learned" the spatial distribution of each class, on a set of training vectors. The Bayesian classification theory provides an ideal method for classification of data, once the a priori probabilities of the classes and their probability density functions are known. The principle of Parzen windows [2] or kernel estimators is thus to estimate the probability density functions with the learning vectors, and then to use these estimates in the Bayes law.

Parzen windows however require a computational load that is unrealistic in practical situations (it requires a.o. the evaluation of a number of Gaussian functions equal to the number of vectors in the learning set); we present here a method to drastically reduce the number of operations involved in Bayesian classification, by using a vector quantization technique to replace the initial

Part of this work has been funded by the ESPRIT project ELENA-Nerves II, supported by the CEC.

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

[†] Fellow of the Belgian IRSIA.

learning set by another one with a strongly reduced number of samples, while minimizing the approximation error on the probability density functions. This method allows to consider favorably the use of kernel estimators in realistic classification tasks.

2. Bayesian classification

Assume the problem consists of classifying an observed vector u of $\mathcal{R}^d$ among c classes denoted ω_j. Assume that u is random and that its d components admit a joint density $p_x(u|\omega_j)$ in class ω_j. If all wrong decisions are given the same penalty, the Bayes law may be expressed as:

$$P(\omega_i|u) = \frac{p_x(u|\omega_i)P(\omega_i)}{\sum_{j=1}^{c} p_x(u|\omega_j)P(\omega_j)}, \tag{1}$$

where $P(\omega_j)$ is the a priori probability of class ω_j, and $P(\omega_i|u)$ the probability that vector u belongs to class ω_i. The Bayesian decision to select the most probable class will thus be:

$$\textit{Decide } u \in \omega_s \Leftrightarrow s = Arg \max_{1 \leq i \leq c} \{P_i \, p_x(u|\omega_i)\}. \tag{2}$$

Using equation 2 necessitates the knowledge of distributions $p_x(u|\omega_i)$ and of a priori probabilities P_i. Given a learning set of vectors, i.e. a set $A_N = \{x(n),\ \omega_{x(n)},\ 1 \leq n \leq N\}$ of vectors $x(n)$ and their associated known classes $\omega_{x(n)}$, it is possible to estimate these distributions and a priori probabilities. The a priori probabilities and simply estimated by the ration between the number of learning vectors in each class and the total number of learning vectors. According to [2], the probability densities in each class can be estimated by

$$\hat{p}_x(N_i, u|\omega_i) = \frac{1}{N_i} \sum_{n=1}^{N_i} K\left(\frac{u - x(n)}{h(n)}\right) \tag{3}$$

where $\{x(n), 1 \leq n \leq N_i\}$ denote the available patterns in a given class ω_i and $K(\cdot)$ a kernel function. The parameter $h(n)$ is called the *width factor* of the kernel, which can either depend of $x(n)$or not. Gaussian kernels are often used:

$$K\left(\frac{u - x(n)}{h(n)}\right) = \frac{1}{\left(h(n)\sqrt{2\pi}\right)^d} \, exp\,(-\frac{1}{2}\left(\frac{||u - x(n)||}{h(n)}\right)^2), \tag{4}$$

where d is the dimension of u and $x(n)$.

The purpose of the following method is to drastically reduce the number of kernels N_i in each class, in order to use equation 3 in realistic situations, avoiding to reduce the quality of the approximation.

3. Suboptimal Bayesian classification

3.1. Principle and hypotheses of the method

The principle of the proposed method is to split the portion of the space where vectors can be found in clusters; a vector quantization technique will be used to find the clusters and their centers of gravity, and it will be assumed that the error generated by the vector quantization will be sufficiently small so that the true probability density inside each cluster can be approximated by a constant. In the portions of the space where the vector quantization will lead to small clusters, this last assumption will be verified; on the other side, in the portions of the space where the clusters are large, this means that the number of learning vectors which lead to these clusters is small, and so that an error in the approximation of the density function is of less importance.

Other algorithms exist to reduce the size of the learning set before using equation 3. The first one [4] extracts a reduced set from the original one in an optimal way to reduce the differences between the probability density estimate before and after this reduction; this method is however heavy on a computational point-of-view, and leads to unsatisfactory results for high reduction rates [8]. Another algorithm [3] uses a vector quantization technique to reduce the size of the learning set, as does our algorithm, but is based on a Gaussian hypothesis of distribution inside each cluster, instead of a constant one for ours; the Gaussian hypothesis is more appropriate when the vector quantization leads to clusters which represent the modes of the distribution, which is the case when the number of clusters is much smaller than in our hypotheses.

3.2. Vector quantization

Any vector quantization method can be used to reduce the size of a dataset. The "Generalized Lloyd Algorithm" [5] is one of the most popular techniques; we will however use an iterative rule known as "competitive learning" or "Kohonen learning rule", based on iterative changes of the codebook, i.e. on a modification of the reduced learning set each time a new learning vector is presented. The iterative character of the rule will be used in the evaluation of the width factors and explained in the next section.

The aim of the Kohonen Learning Algorithm is to approximate the sets of patterns A_{N_i} by sets of so-called centroids $B_{M_i} = \{c(m), \omega_{c(m)} = \omega_i, 1 \leq m \leq M_i\}$, where $M_i << N_i$, roughly keeping the same probability density of vectors for sets A_{N_i} and B_{M_i}. The principle of the KLA method is then the following in each class ω_i.

First, the M_i centroids $c(m)$ are randomly initialized to any of the N_i patterns, keeping the same a priori probabilities of classes for both sets A_{N_i} and B_{M_i}. Then, each of the N_i patterns $x(n)$ is presented to the set B_{M_i}, and the centroid $c(a)$ closest from $x(n)$ is then selected and moved in the direction of the presented pattern :

$$c(a) = c(a) + \alpha(x(n) - c(a)) \tag{5}$$

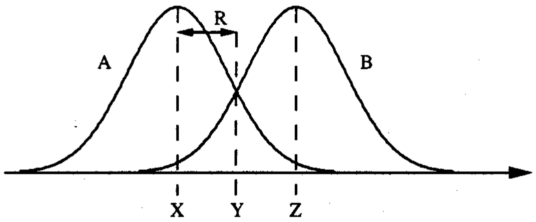

Figure 1: Illustration of two Gaussian functions in 1 dimension, and related notations.

where α is an adaptation factor ($0 \leq \alpha \leq 1$) which must decrease with time during the learning to ensure the convergence of the algorithm. After several presentations of the whole set of patterns A_{N_i}, the distribution of centroids $c(m)$ in B_{M_i} will reflect this of the pattern set A_{N_i}.

For the estimation of probability densities in each class, we will use the reduced sets B_{M_i} instead of the original sets A_{N_i}; this will strongly decrease the number of operations involved in 3.

3.3. Width factors

Our hypothesis is that the true probability density can be considered as constant inside each cluster; this leads to the constraint that we will choose the width factor of the Gaussian function associated to each cluster in order to keep the estimate 3 of the density as constant as possible over two consecutive clusters. Let us examine the one-dimensional example of figure 1. X and Z represent the centers of two consecutive clusters A and B, $2R$ the distance and Y the midpoint between them. The purpose of the method is to set the relation between R and the width factor h of the Gaussian functions A and B, in order to have a constant approximate of the probability density over the segment $[X, Z]$. We will simplify the computation of h by setting its value in order to have the same estimate of probability density at points X, Y and Z; we assume that the fluctuations inside the segments $[X, Y]$ and $[Y, Z]$ may be neglected. We will also neglect the influence of a kernel at a distance $2R$ of its center (which, when verified a posteriori, will cause a maximum error of about 6% in the local value of the estimate).

With these hypotheses, we can evaluate the contribution of the Gaussian functions A and B respectively at locations X (or Z) and Y:

$$\hat{p}_x(\{A, B\}, X) = \frac{1}{h(m)\sqrt{2\pi}} \tag{6}$$

$$\hat{p}_x(\{A, B\}, Y) = \frac{2}{h(m)\sqrt{2\pi}} \mathrm{e}^{\frac{-R^2}{2h(n)^2}} \tag{7}$$

Making the estimates 6 and 7 equal to have an equal approximation of probability density at points X, Y and Z leads then to

$$R = \sqrt{2 \ln 2}\, h(m) \tag{8}$$

A similar development may be done in dimension 2. In this case, we can consider the approximation of probability density due to four Gaussian functions A, B, C and D centered on the four vertices of a square; this approximation is computed at the location X of a vertex of the square, at the center Y of the square, and at the midpoint Z of an edge. Setting the distance between two centroids on an edge of the square equal to $2R$, and neglecting the influence of kernels at a distance greater or equal to $2R$, we have respectively:

$$\hat{p}_x(\{A,B,C,D\},X) = \frac{1}{\left(\sqrt{2\pi}\, h(m)\right)^2} \tag{9}$$

$$\hat{p}_x(\{A,B,C,D\},Y) = \frac{4}{\left(\sqrt{2\pi}\, h(m)\right)^2} \mathrm{e}^{\frac{-R^2}{h(m)^2}} \tag{10}$$

$$\hat{p}_x(\{A,B,C,D\},Z) = \frac{2}{\left(\sqrt{2\pi}\, h(m)\right)^2} \mathrm{e}^{\frac{-R^2}{2\,h(m)^2}} \tag{11}$$

It is possible to make the estimations 9, 10 and 11 equal, by setting the width factor $h(m)$ according to equation 8, which gives thus the same result as in dimension 1. An identical development can be made in dimension 3, by considering the influence of 8 Gaussian kernels located on the vertices of a cube, respectively at the locations of these vertices, of the midpoint of any edge, of the center of a face, and of the center of the cube. Again, the estimations of probability densities at these points will be equal if equation 8 is respected; it will also be the case in dimension d greater than 3.

Now we have the relation between $h(m)$ and R, we need a method to evaluate R. First, we will evaluate the inertia of each cluster, by using an adaptive method exactly as the competitive learning does for the locations of the centers. The *inertia* coefficient $i(m)$ for each cluster is computed in the following way:

$$i(a) = i(a) + \alpha(||x(n) - c(a)||^2 - i(a)) \tag{12}$$

where a is the index of the closest centroid to a learning vector $x(n)$. Equation 12 is a kind of convex combination at each iteration between the previously estimated value of $i(a)$ and a new contribution $||x(n) - c(a)||^2$ due to the input vector $x(n)$. After learning, parameters $i(m)$, $1 \leq m \leq M_i$, will converge to the average inertia of points in the clusters associated to $c(m)$.

The last point to solve is the relation between the estimated inertia $i(a)$ and the distance R. If we consider that, under the locally uniform density approximation as above, the local arrangement of the centers of consecutive clusters will be as the vertices of an hypercube with edges of length $2R$, the relation between the inertia of each cluster and R is:

$$i(m) = \frac{1}{(2R)^d} \int_V ||x(n) - c(m)||^2 dV = \frac{dR^2}{3} \tag{13}$$

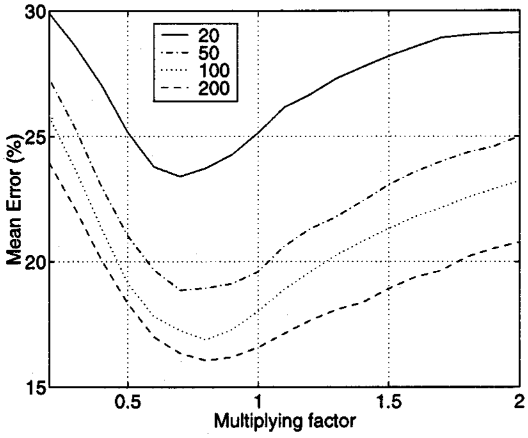

Figure 2: Percentage of classification error on the phoneme database with 20, 50, 100 and 200 centroids.

Combining equations 8 and 13 then leads to a width factor $h(n)$ given, in dimension d, by

$$h(n) = \sqrt{\frac{3\,i(n)}{2\,d\ln 2}} \tag{14}$$

Finally, the estimation of probability density in each class will be calculated through equation 3, applied on a set of centroids fixed by 5, the width of the kernels being fixed by 14. Bayesian classification is then realized through equation 2, where the probability densities are replaced by the above estimates, and the a priori probabilities by percentage of occurrence of prototypes $x(n)$ in each class. This constitutes the IRVQ (Inertia-Rated Vector Quantization) method.

4. Simulation results

Simulations have been carried out on two real-world classification databases.

The first one, "phoneme" was in use in the European ROARS ESPRIT project [1]. It's aim is to distinguish between the classes of nasal and oral vowels. The database contains 5427 vowels coming from isolated syllables (for example: pa, ta, pan,...). Five different attributes characterize each vowel: the amplitudes of the five first harmonics, normalised by the total energy (integrated on all the frequencies).

The second database, "satimage" comes from the ftp anonymous "*UCI Repository Of Machine Learning Databases and Domain Theories*" [6]. It was in use in the European STATLOG ESPRIT project [7]. This database was generated from Landsat Multi-Spectral Scanner image data purchased from NASA by the Australian Centre for Remote Sensing. It is a (tiny) sub-area

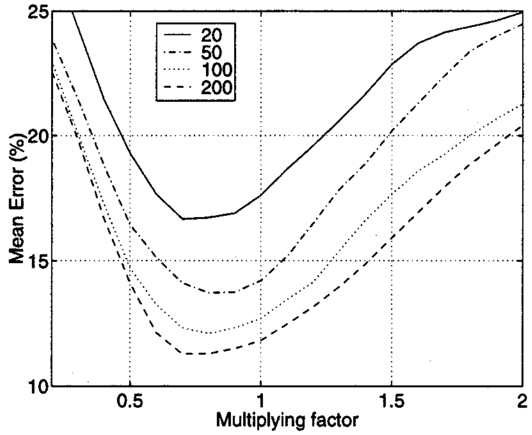

Figure 3: Percentage of classification error on the satimage database with 20, 50, 100 and 200 centroids.

of a scene, consisting of 82 x 100 pixels. Each line of data corresponds to a 3x3 square neighbourhood of pixels completely contained within the 82x100 sub-area. Each line contains the pixel values in the four spectral bands of each of the 9 pixels in the 3x3 neighbourhood and a number indicating one of the 6 classification labels of the central pixel (red soil, cotton crop, grey soil, ...). The database contains 6435 patterns with 36 attributes (4 spectral bands x 9 pixels in neighbourhood).

The test used in these two cases was the holdout method averaged on five partitions of the original database in two independant learnset and testset containing each the half of the total amount of available patterns. Simulations consisted in measuring the error percentage on the testsets of the Bayesian classifier built with the estimations of probability densities on the learnset, after vector quantization leading to a total number of 20, 50, 100 or 200 clusters (for all classes together). In order to evaluate the correctness of equation 14, the width factors $h(n)$ have been multiplied by a factor varying from 0 to 2; the value of 1 corresponds thus to equation 14.

Figures 2 and 3 clearly show a minimum in the value of the error; this minimum is however obtained for a multiplying factor slightly inferior to 1, i.e. for a width factor slightly inferior to our evaluation 14. This small difference is due to the numerical approximations made in the derivation of 14. Simulations carried out on other databases showed similar results.

5. Conclusion and future work

Efficient computing of probability densities for Bayesian classification requires sub-optimal methods, avoiding to compute as many kernel functions as there are vectors in the learning set. Vector quantization techniques have been shown to reach this goal. The challenge is however to evaluate appropriate widths for the kernels used in the estimation of probability densities; based on an hypothesis of small clusters, i.e. of constant probability densities over two conscutive clusters, we derived a theoretical value for the widths, depending on the measured inertia of the clusters.

Extensive experiments showed an optimum value of the widths slightly inferior to our theoretical value; the small difference is due to the numerical approximations made in the development.

While the vector quantization process is deemed to have converged to clusters having the same distribution as the initial points, simulations showed that this process is often trapped in local minima, leading to clusters including different number of points of the initial database. Future work consists in taking into account the number of points in each cluster to increase the quality of approximation.

References

[1] P. Alinat. Periodic Progress Report 4. Technical report, ROARS Project ESPRIT II- Number 5516, February 1993. Thomson report TS. ASM 93/S/EGS/NC/079.

[2] T. Cacoullos. Estimation of a multivariate density. *Annals of Inst. Stat. Math.*, 18:178–189, 1966.

[3] P. Comon, G. Bienvenu, and T. Lefebvre. Supervised design of optimal receivers. In *NATO Advanced Study Institute on Acoustic Signal Processing and Ocean Exploration*, Madeira, Portugal, July 26-Aug. 7 1992.

[4] K. Fukunaga and R.R. Hayes. The reduced Parzen classifier. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(4):423–425, April 1989.

[5] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, January 1980.

[6] 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).

[7] G. Nakhaeizadeh. Project STATLOG . Technical report, ESPRIT IPSS-2 Number 5170 : Comparative Testing and Evaluation of Statistical and Logical Learning Algorithms for Large-Scale Applications in Classification, Prediction and Control, April 1993.

[8] Q. Xie, C. A. Laszlo, and R. K. Ward. Vector quantization technique for nonparametric classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1326–1330, december 1993.