"Les réseaux neuromimétiques et leurs applications"
Marseille, 15-16 déc. 1994.

94-C

HANDWRITTEN DIGIT RECOGNITION BY SUBOPTIMAL BAYESIAN CLASSIFIER

Jean-Luc Voz, Michel Verleysen, Philippe Thissen & Jean-Didier Legat

Université Catholique de Louvain
Microelectronics Laboratory
3 place du Levant
1348 Louvain-La-Neuve
Belgium

Tel.: +32 10 47 25 51 - Telefax: +32 10 47 86 67
E-mail: voz@dice.ucl.ac.be

Abstract

Non-parametric kernel estimation of probability densities provides a useful way to build Bayesian classifiers that may be used, for example, in optical character recognition problems. The complexity of conventional kernel estimators is however far beyond the acceptable limits for performant systems. We present, in this paper, a neural-like vector quantization technique, the Inertia-Rated Vector Quantization (IRVQ), which allows to strongly decrease the complexity of kernel estimators. We apply this original technique to the recognition of handwritten numerals and we prove its interest through high recognition rates coupled to low memory and computational requirements.

Keywords

Bayesian classification, kernel estimators, vector quantization, optical character recognition

Résumé

Les estimateurs à noyaux de densités de probabilités permettent de réaliser des classificateurs Bayésien pouvant, par exemple, être utilisés pour des problèmes de reconnaissance optique de caractères. La complexité des estimateurs à noyaux classiques reste néanmoins loin des limites acceptables pour la réalisation de systèmes performants. Cette communication présente une méthode neuromimétique de quantification vectorielle, « l'Inertia-Rated Vector Quantization » (IRVQ), permettant de décroître sensiblement la complexité des estimateurs à noyaux. Cette technique originale appliquée à la reconnaissance de chiffres manuscrits nous a permis d'obtenir des taux élevés de reconnaissance associés à des besoins en mémoire et en temps de calcul extrêmement faibles.

Mots clés

classification Bayésienne, estimateurs à noyaux, quantification vertorielle, reconnaissance optique de caractères

1 Introduction

Classification of high-dimensional data is a challenging task in engineering science. The Bayes theory provides the mathematical tools to study classification problems; however, to minimize the number of misclassifications in a problem according to the Bayes law, the knowledge of the probability densities in each class is necessary. Estimations of these probability densities may be obtained through kernel estimators, or Parzen windows [1, 2, 3]; however, such methods involve tremendous high numbers of computations, and are thus inefficient in most practical applications.

We present here a vector quantization technique to drastically reduce the number of operations involved in probability densities estimation, by selecting a limited number of points representing the training distribution, and estimating the densities from these points only. This method is applied to the recognition of writer-independent handwritten numerals, and is tested on a database of handwritten digits. The preprocessing used before the classifier is a classical extraction of topological characteristics; the test results show very good performances of the classifier itself. The advantages of our method comes from the adaptive training as in neural networks, and also from the limited numbers of computations as required in real-time OCR systems.

2 Bayesian classification

2.1 Nonparametric Bayesian classifier design

Assume the problem consists of classifying an observed vector x of $\mathcal{R}^d$ among C classes denoted ω_j. Assume that x is random and that its d components admit a joint density $p_x(u)$. If all wrong decisions are given the same penalty, the Bayesian decision will be:

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

where $p_x(u|\omega_i)$ is the density of the vector x under the hypothesis that it belongs to class ω_i and P_i is the a priory probability that class ω_i occurs. This corresponds to attribute u to the class which has the greatest a posteriori probability $P(\omega_i|u)$.

It is quite obvious that such an ideal Bayesian solution can be used only if distributions $p_x(u|\omega_i)$, and the C a priori probabilities P_i are known. In the problems we are interested in, it is seldom the case. We rather have at disposal a set of patterns, $A_N = \{x(n),\ \omega_{x(n)},\ 1 \le n \le N\}$, where each pattern $x(n)$ belongs to a known class $\omega_{x(n)}$. Denote N_i the number of available patterns in class $\omega_i, 1 \le i \le C$, $\sum_{i=1}^{C} N_i = N$, and $A_{N_i} = \{x(n) \mid \omega_{x(n)} = \omega_i,\ 1 \le n \le N_i\}$.

The set A_N may be used to estimate the parameters of classical linear or quadratic classifiers, but when the underlying densities $p_x(u|\omega_i)$ are non-Gaussian (which is mostly the case in real-world problems), the error of these popularly used parametric classification approaches is frequently much larger than the error obtained by nonparametric techniques. One way of performing nonparametric Bayesian classifier design 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. Kernel estimators [1, 2, 4] provide a useful way to estimate probability densities. The kernel estimate of the density $p_x(u|\omega_i)$ of a random variable x takes the following general form:

$$\hat{p}_x(N_i, u|\omega_i) = \frac{1}{N_i} \sum_{n=1}^{N_i} \frac{1}{h(n)^d} K\left(\frac{u - x(n)}{h(n)}\right) \tag{2}$$

where $\{x(n), 1 \le n \le N_i\}$ 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. If $h(n)$ 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 $x(n)$. Better estimates are always obtained with variable kernel width factors, but an important problem is to obtain their optimal values. In [3] an iterative method (the RRE algorithm) is proposed to compute the optimal variable $h(n)$ from the design set.

Several types of radial kernels $K(\cdot)$ may be used, the most classical one being a Gaussian function:

$$K\left(\frac{u-x(n)}{h(n)}\right) = \frac{1}{\left(\sqrt{2\pi}\right)^d} \; exp\left(-\frac{1}{2}\left(\frac{\|u-x(n)\|}{h(n)}\right)^2\right), \tag{3}$$

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

2.2 Suboptimal Bayesian classifier

In nonparametric kernel approaches, the obtaining of adequate statistical information always requires a large number N of prototypes in the design set. This implies the need of very large amount of computer storage and the number of operations involved in the evaluation of equation (2) may lead to unacceptable computation times for on-line applications like OCR. The solution to this problem is to build the kernel classifier from a reduced design set, chosen to keep the performances as near as possible from those of the classifier built on the entire original learning set.

Two reduction methods were proposed in this sense. The first data reduction algorithm developed by Fukunaga and Hayes in [5] extract from the original design set an "optimal" reduced set in the sense that the difference between the probability density function estimated from the reduced set and that estimated from the original one is minimized. But this method has high computational burden and unsatisfactory classification results for high reduction rate [6]. The second idea [4, 6], which is used in this paper, is to apply vector quantization techniques to build the reduced design set, using the statistical information of the original one during the quantization process. A justification of the fact that the reproduction vectors obtained by an optimal quantizer could be used as an effective design set to build the suboptimal Bayesian classifier may be found in [6].

The optimal locations of the reduced classifier's kernels being set by a quantization technique, we still have to compute their optimal variable width factors. The RRE method proposed in [3] cannot be used in our case since the assumption of an infinity tending design set is not more valid for the reduced design set. The CAK algorithm, proposed in [7] and inspired by [4] assimilates the density in each cluster to isotropic gaussian. Empirical studies of this algorithm let us think that this assumption is not valid in most of the real-world problems. Our new iterative method, detailed in the next section will be based on a significantly different assumption.

2.3 The IRVQ algorithm

Vector quantization provide a useful way to reduce the number of high-dimensional vectors in a set, while keeping their distribution unchanged [8]. The Generalized Lloyd Algorithm (GLA) [9] is the most widely used one in various industrial problems such as image compression. A neural network "on-line" method ("Competitive Learning", or "Kohonen Learning Algorithm (KLA)") which provides an interesting computationally efficient substitute to the GLA algorithm for comparable overall performances may be found in [10].

The purpose of the Kohonen Learning Algorithm is to approximate the sets of patterns A_{N_i} by sets of so-called centroids $B_{M_i} = \{c(m), \omega_{c(m)} = \omega_i, 1 \leq m \leq M\}$, where $M_i << N_i$, and roughly keeping the same probability density of vectors in the space for the 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}; the centroid $c(a)$ closest from the presented pattern $x(n)$ is moved in the direction of $x(n)$:

$$c(a) = c(a) + \alpha(x(n) - c(a)) \tag{4}$$

where α is an adaptation factor ($0 \leq \alpha \leq 1$) which must decrease with time 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}.

During the adaptation process (equation 4), it is possible to keep a trace of the mean distance between a pattern $x(n)$ and its closest prototype $c(a)$, by affixing an *inertia* coefficient $i(m)$ to each centroid $c(m)$, $1 \leq m \leq M_i$. This inertia coefficient is randomly initialized to a small value and then adapted at the same time as the centroid locations (equation 4) according to:

$$i(a) = i(a) + \alpha(\|x(n) - c(a)\|^2 - i(a)) \tag{5}$$

After learning, parameters $i(m)$, $1 \leq m \leq M_i$, will converge to the average inertia of points in the clusters associated to $c(m)$. We will use this inertia coefficient $i(m)$ for the computation of the optimal variable width factor associated to the kernel centered on $c(m)$ in the reduced classifier.

We first suppose that the true density $p_x(u|\omega_i)$ in each class may be approximated by a constant value inside each cluster defined by the Voronoi tessellation obtained from the quantized codebook. We then suppose that a constant probability density over two consecutive clusters will be well approximated if the value of the probability density estimate $\hat{p}_x(N_i, u|\omega_i)$ (the sum of the contributing kernels) is identical on the location of the centroids themselves and on the borders between two clusters in the Voronoi tessellation.

After neglecting the contribution of distant kernels, we find that the width $h(m)$ of a specific kernel must be

$$h(m) = \frac{R}{\sqrt{2\,ln2}} \tag{6}$$

to have a constant approximation of probability density inside the associated cluster, where ln is the natural logarithm and $2R$ is the distance between two consecutive centroids.

The relation between the estimated inertia $i(m)$ and R can then be computed by supposing that the cluster may be approximated by a hypercube with edges of length $2R$:

$$i(m) = \frac{dR^2}{3} \tag{7}$$

Equation (6) and (7) then give the relation between the estimated inertia and the width factor of the kernel:

$$h(m) = \sqrt{\frac{3\,i(m)}{2\,d\,ln2}} \tag{8}$$

This original method, coupling the KLA with the computation of inertia coefficients (equation 5), is called IRVQ, for Inertia-Rated Vector Quantization. Figure 1 shows the results of the IRVQ algorithm on three different two-class problems, the first one for gaussian mixture distributions, the second one with normal distributions, and the last one for uniform concentric circular distributions. The IRVQ algorithm is then used as an initialization step for the kernel-based estimator of the probability densities (equation 2) : however, the sum on all N_i patterns $x(n)$ is now replaced by a sum on the M_i centroids $c(m)$, and the width factors $h(m)$ are set by equation (8). The final classifier is then built according to equation (1) with the estimates of probability densities so computed and the estimates of the priors P_i given by the proportion of points in each class.

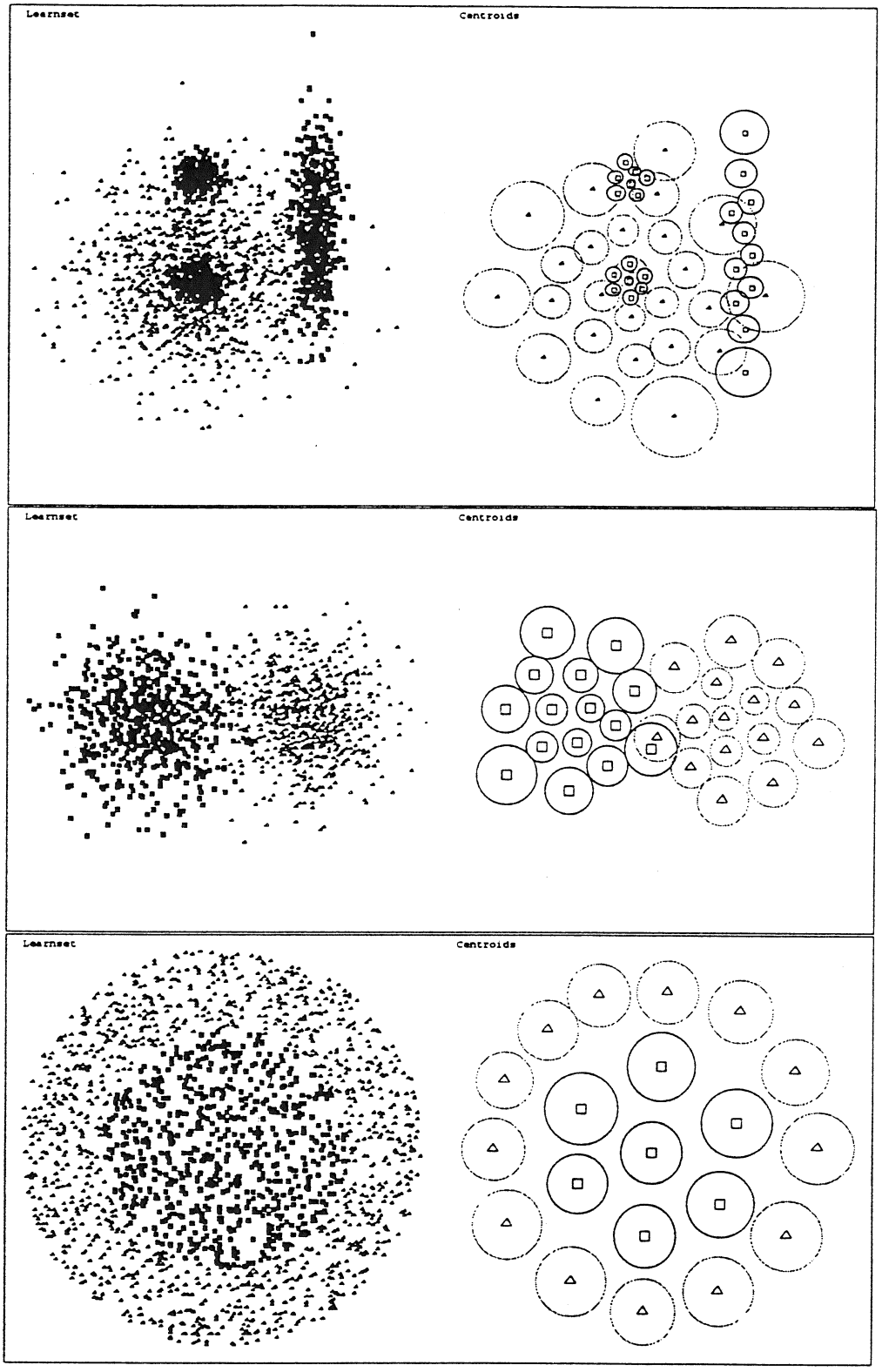

Figure 1: Centroids and their associated optimal width factors obtained by the IRVQ algorithm for three different two-class bidimensional databases.

3 Application to Optical Character Recognition

3.1 Database presentation

A number of simulations have been carried out on a real-world dataset provided by the ATT Bell Laboratories [11, 12]. It consists of 1200 handwritten numerals, containing 120 examples of each digit, written by 12 different people. Each person wrote 10 times the same succession of numerals from 0 to 9. Each data is a centered bitmap normalized to the size 16x16 pixels. Figure 2 shows the pixel map representation after normalization of three series of digits written by three different people who had been asked to follow a given writing style (the distributions are thus supposed to be monomodal in each class). Among the 10 examples of each digit written by each person, the first 5 examples were placed in the training set, and the remaining 5 were placed in the test set. The test conditions were thus the same than in [11].

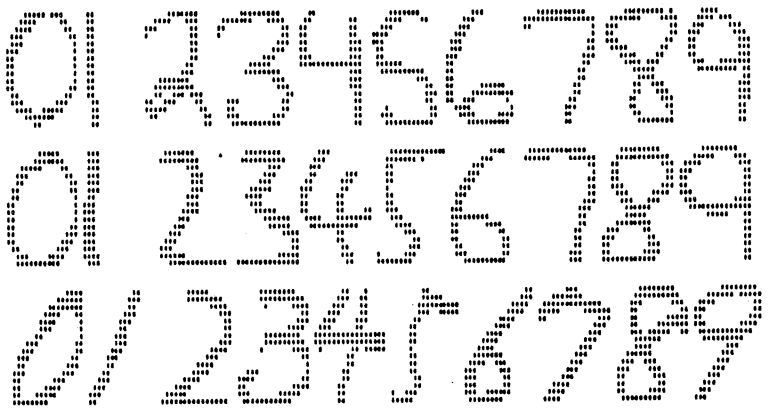

Figure 2: Examples of normalized handwritten digits from the database.

3.2 Features extraction and preprocessing

The preprocessing described here is a general method applicable in most cases of handwriting recognition. Its use on the Bell handwritten digit database combined with the IRVQ method allowed us to obtain the highest recognition results.

The problems encountered by a handwritten recognition system are important : variation of sizes, presence of defaced characters, similar shapes,... Topological features are well adapted to size variation and makes possible the recognition of handwritten characters with very few constraints. Topological recognition techniques trying to represent the general "concept of the character" based on a general description of its structure have been investigated by many researchers [13].

Each character is described after preprocessing by a feature vector $f = (f_1, f_2, f_3, ..f_{15})$ whose elements are topological information about the character. The features have been designed in order to minimize their number, and so to reduce the memory size needed by the classifier.

Topological Features are extracted from the binary image. The character is described in term of "bars" and "holes". Furthermore, the approximate position of these bars and holes is also used. The bitmap image is scanned in four directions (top to bottom, bottom to top, left to right and right to left) and projections, outlines and stroke densities vectors are calculated. These features are illustrated on figure 3 where the top profile and the vertical histogram are represented for the machine-printed character "M". These are typical low-level features that can be calculated along various directions [14]. For a typical 16 x 16 image, 12

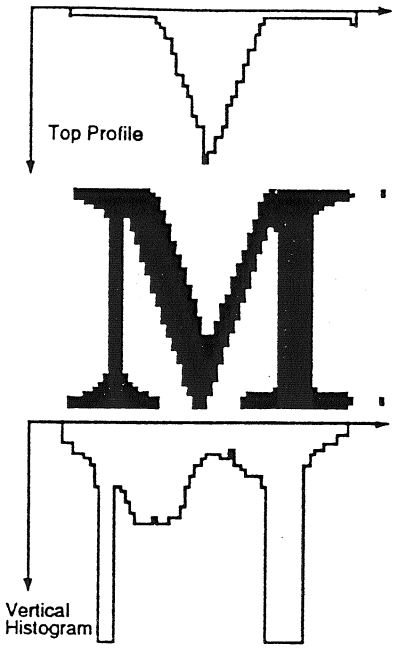

Figure 3: Two topological features of the "M" character.

features vectors with a total of 192 elements are generated. These basic features are then combined to extract the bars and holes present in the image of the character.

The topological feature vector f contains 15 elements and has a fixed size :

- f_0 represents the aspect ratio of the character,
- f_1 to f_3 represent upper holes (left, middle, right),
- f_4 to f_6 represent lower holes (left, middle, right),
- f_7 to f_9 represent left holes (top, middle-up, middle-down),
- f_{10} to f_{12} represent right holes (top, middle-up, middle-down),
- f_{13} to f_{14} represent horizontal bars (top, bottom).

Each of the first 15 elements can be present or absent. The absence of a feature is represented by a "0" value (for instance, a character like a 7 has no upper hole). If the feature exists, its intensity is calculated. The intensity of a feature represents its importance (small, large, thin, wide, etc...) regarding the global shape of the character. The intensity of a feature is then normalized and doesn't depend on the size of the image representing the character. The intensity of a feature is calculated on a scale ranging from 0 (the feature doesn't exist) to 15 (very strong feature). The aspect ratio of the character is the only element of the feature vector where the relative value of the width and the height are taken into account. The feature vector (except for the aspect ratio) doesn't depend on the size of the character, i.e. it is identical if an horizontal and/or vertical scaling of the character is done except for discrete calculation effects. It is then well adapted to the recognition of handwritten characters in any size.

3.3 Recognition results

Three approaches are used in order to study the performances of the preprocessing and classification methods detailled in this paper:

- First, we compare the performances of our preprocessing method with the one presented in [11], using in both cases a standard Nearest Neighbour classifier; the percentages of correct classifications are respectively 98.3% and 96.3%, which shows the well-founded of our feature extraction method.
  The memory size needed to store one preprocessed pattern has been computed by the same method than in [11]: 60 bits (15 features which must be coded on 4 bits). This is comparable to the value reported in [11] for their method: 99 bits.

- Secondly, to analyse the performances of the IRVQ classifier independantly of the preprocessing, we compare its results with the performances of other classical neural network classifiers build to require comparable memory sizes:

  - **IRVQ:** The training of the IRVQ algorithm was performed by presenting 20 times the 600 preprocessed training samples of the Bell database (20 epochs), using 5 centroids per class, the α adaptation factor linearly decreasing from 0.3 to 0 during the learning.
    Required memory size: 60 bits per centroids multiplied by their number (50) and summed with the memory size needed to store the width factors of each centroid (50x8 bits): 3400 bits.
    Table 1 shows the confusion matrix on the 600 samples of the test set. Each line of the confusion matrix represents a class; the percentage of elements from this class attributed to any class is given in each of the elements of the line.

Class	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	96.6	0.0	0.0	0.0	0.0	0.0	1.7	0.0	1.7	0.0
1	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	1.7	98.3	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	95.0	0.0	0.0	0.0	3.3	1.7
5	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	1.7	0.0	98.3	0.0	0.0
8	1.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.3	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0

Table 1: Confusion matrix.

  - **NN:** A Nearest Neighbour classifier build on a reduced design set of 50 patterns obtained by a k_means learning algorithm trained until convergence.
    Required memory size: 60 bits per stored pattern multiplied by their number (50): 3000 bits.
  - **LVQ:** A Learning Vector Quantizer classifier [15] designed with a codebook of 50 patterns trained in the same contions than for the IRVQ.
    Required memory size: 60 bits per stored pattern multiplied by their number (50): 3000 bits.
  - **RCE:** A Reduced Coulomb Energy classifier [16] trained to reach a 100% recognition rate on the learning set, the initial radius being set to 5. This learning generated 40 centroids.
    Required memory size: 60 bits per centroids multiplied by their number (40) and summed with the memory size needed to store the width factors of each centroid (40x8 bits): 2720 bits.

  Table 2 compares the performances and estimated memory sizes of the IRVQ classifier with the values obtained for the classifiers here above, trained on the Bell database preprocessed by the feature extraction method explained in the previous section. The IRVQ classifier provides the best results for comparable memory size.

Method	Performance	Size
IRVQ	98.7	3 400
NN	97.0	3 000
LVQ	94.7	3 000
RCE	93.7	2 720

Table 2: Performances and sizes of the IRVQ classifier in comparison with the NN, LVQ and RCE classifiers for the Bell database preprocessed by the same method.

- Finally, we compare the results of our complete classification chain (preprocessing and classifier) with the best one reported in [11]: a specific feature extraction method associated with a 2 by 2 class separation neural network algorithm. The values in table 3 illustrates the excellent performances and low-cost properties of the preprocessing and classification methods detailled in this paper.

Method	Performance	Size
IRVQ	98.7	3 400
Best of [11]	98.0	29 537

Table 3: Performances and sizes of the IRVQ classifier associated to the feature extraction method developped in this paper in comparison with the best classifier reported in [9].

4 Conclusion and future works

The use of vector quantization techniques allows to limit the sizes of databases to reasonable values. We showed in this paper how to extend these techniques to serve as preprocessing to Bayesian kernel classifiers using estimates of in-class probability densities. We applied this technique to the recognition of handwritten numerals, and found high recognition rates coupled to low requirements in terms of memory and computational resources. The results are encouraging for the use of the technique presented in this paper to more complex OCR problems, our aim being to apply this method to the problem of on-line cursive character recognition. A performant adaptive vector quantization method such as the one presented in [17] could also be used for this purpose. The classification results obtained with the IRVQ method on several databases lead us to study its implementation on a fully parallel mixed analog-digital architecture [18].

5 Acknowledgments

Part of this work has been funded by the ESPRIT-BRA project 6891, ELENA-Nerves II, supported by the Commission of the European Communities (DG XIII). Michel Verleysen is a Senior Research Assistant of Belgian National Fund for Scientific Research (FNRS). Philippe Thissen is working towards the Ph.D. degree in microelectronics under an IRSIA (Institut pour l'Encouragement de la Recherche Scientifique dans l'Industrie et l'Agriculture) fellowship.

References

[1] T. Cacoullos, "Estimation of a multivariate density", *Annals of Inst. Stat. Math.*, vol. 18, pp. 178–189, 1966.

[2] E. Parzen, "On the estimation of a probability density function and the mode", *Ann. Math. Stat.*, vol. 27, pp. 1065–1076, 1962.

[3] P. Comon, J.L. Voz, and M. Verleysen, "Estimation of performance bounds in supervised classification", in *ESANN94-European Symposium on Artificial Neural Networks*, M. Verleysen, Ed., Brussels, Belgium, April 1994, pp. 37–42, D facto publications.

[4] P. Comon, "Classification bayésienne distribuée", *Revue Technique Thomson CSF*, vol. 22, no. 4, pp. 543–561, 1990.

[5] K. Fukunaga and R.R. Hayes, "The reduced Parzen classifier", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 4, pp. 423–425, Apr. 1989.

[6] Q. Xie, C. A. Laszlo, and R. K. Ward, "Vector quantization technique for nonparametric classifier design", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 12, pp. 1326–1330, december 1993.

[7] 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.

[8] A. Gersho, "Asymptotically optimal block quantization", *IEEE Transactions on Information Theory*, vol. IT-25, pp. 373–80, July 1979.

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

[10] E. Yair, K. Zeger, and A. Gersho, "Competitive learning and soft competition for vector quantizer design", *IEEE Transctions on Signal Processing*, vol. 40, no. 2, pp. 294–309, 1992.

[11] I. Guyon et al., "Comparing different neural architectures for classifying handwritten digits", in *IJCNN89*, Whashington, 1989.

[12] L.D. Jackel et al., "An application of neural net chips : Handwritten digits", in *IEEE Int. Conf. on Neural Nets*, San Diego, July 1988, pp. 107–115.

[13] C. Y. Suen, "Distinctive features in automatic recognition of handprinted characters", *Signal Processing*, vol. 4, pp. 193–207, 1982.

[14] P. De Muelenaere, M. Dauw, and J.D. Legat, "Omnifont recognition using topological recognition techniques", in *11th IAPR , Int. Conf. on Pattern Recognition*, The Hague, Sepember 1992, pp. 410–413, vol. B.

[15] T. Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, Berlin, 1989, 3rd Edition.

[16] D.L. Reilly, L.N. Cooper, and C. Elbaum, "A neural model for category learning", *Biological Cybernetics*, vol. 45, no. 1, pp. 35–41, 1982.

[17] P. Demartines and J. Hérault, "Representation of nonlinear data structures through a fast VQP neural networ3", in *NeuroNîmes93 (Neural Networks and their applications)*, Paris, France, October 1994, pp. 411–424, EC2.

[18] M. Verleysen, P. Thissen, J.L. Voz, and J. Madrenas, "An analog processor architecture for neural network classifier", *IEEE Micro*, vol. 14, no. 3, pp. 16–28, June 1994.