ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 209-214

Optimal decision surfaces in LVQ1 classification of patterns

Michel Verleysen, Philippe Thissen, Jean-Didier Legat

Catholic University of Louvain
Microelectronics Laboratory
3, pl. du Levant
B-1348 Louvain-la-Neuve
Belgium

Abstract. Kohonen's LVQ1 procedure is widely used for the classification of patterns in a multi-class distribution. This algorithm approximates the probability densities of vectors in each class, by updating reference vectors at each presentation of an input pattern. It will however be shown in this paper that the Bayes classification is only approximated by the decision surfaces of LVQ1 in special cases of probability densities; we will show how to set the decision surfaces in a more general case, and how the relative number of reference vectors in each class can comply with their a priori probabilities, condition which is necessary for a good classification.

1. Introduction

LVQ algorithms [1] are used in data processing for sub-space quantization and for the classification of patterns. The principle of LVQ1 is to randomly choose an initial set of reference vectors ('prototypes'), and to sequentially adapt these prototypes in order to approximate first the intra-class stimuli distribution, and secondly the Bayes decision surfaces between classes, to minimize the number of misclassifications. Several problems however occur. First, Kohonen claims that the initial proportion of vectors among the classes must comply with their a priori probabilities; this is true, but not sufficient to ensure good approximations of the Bayes decision surfaces. Furthermore, a nearest-neighbour method is not optimum to approximate the boundaries between classes; the Bayesian criterion indeed requires to set the boundaries at the intersection of the probability densities, and not on the mid-point between two prototypes from different classes. These concepts will be detailed in sections 2 and 3 for 1-dimensional distributions; the results can easily be extended to multi-dimentional stimuli spaces.

2. Learning Vector Quantization (LVQ1)

The LVQ1 algorithm can be described as follows.

Before the first iteration, P K-dimensional prototypes p(i) (i=1...P) are randomly initialized. If a priori limits of the set of input vectors in the space are known, the prototypes will be chosen inside these limits, in order to accelerate the convergence of the algorithm. One possibility is to initialize the prototypes to any P of the N input vectors. At each iteration, a K-dimensional input vector x(j) (j=1...N) is compared to

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 209-214

all prototypes. We then select the prototype p(a) for which the standard Euclidean distance between p(a) and x(j) is minimum

$$\text{dist}\big(p(a), x(j)\big) \leq \text{dist}\big(p(k), x(j)\big) \qquad \forall k \in \{1...P\} \setminus \{a\} \tag{1}$$

($1 \leq a \leq P$). If vectors x(j) and p(a) belong to the same class, p(a) is moved in the direction of x(j) following equation (2).

$$p(a) = p(a) + \alpha \big(x(j) - p(a)\big) \tag{2}$$

where α is the adaptation factor ($0 < \alpha < 1$). If the two vectors belong to different classes, p(a) is moved in the opposite direction of x(j):

$$p(a) = p(a) - \alpha \big(x(j) - p(a)\big) \tag{3}$$

The adaptation factor α must decrease with time to obtain a good convergence of the algorithm [2]. Usually, the same value of the adaptation factor α is kept for a whole 'epoch' (an epoch consists in presenting once the whole set of input patterns), and is decreased before the next one.

3. Variable influence regions at boundaries between classes

For the sake of simplicity, let us examine the one-dimensional case of figure 1, where two classes overlap, with linear probability densities in each class. The following of this section is valid for probability densities that can be linearly approximated; this condition is respected when the number of prototypes is large enough to obtain a good approximation of the probability densities by a linearization within the influence region of each prototype. If we assume that the two classes have equal a priori probabilities, and that the number of prototypes in each class is constant, a 1-dimensional LVQ1 algorithm applied to the illustrated distributions will lead to the two sets of centroïds respectively marked by x and o.

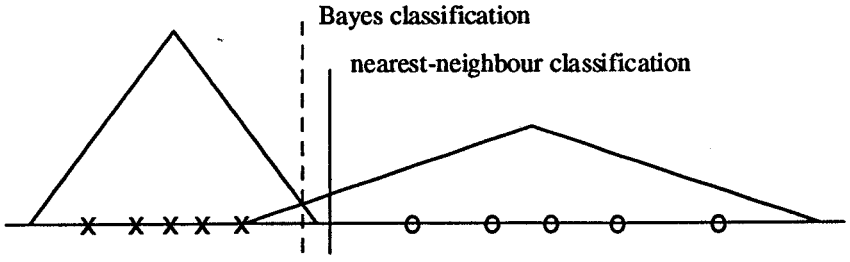

Fig. 1. LVQ1 algorithm on 1-D overlapping triangular distributions

The vertical plain line in figure 1 illustrates the nearest-neighbour criterion of equation (1). The dotted line shows the ideal Bayes boundary, minimizing the number of misclassifications; it is obvious that a more appropriate choice of the boundaries, instead of criterion (1), could lead to a better classification.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 209-214

3.1. Approximated difference of probability densities

Equations (2) and (3) indicate that prototypes are moved in the direction of the stimuli if they belong to the same class, and are moved in the opposite direction if they belong to different classes. In figure 2, we first suppose that a nearest-neighbour criterion is not used to determine the boundaries between classes, but that another appropriate criterion makes it possible to separate the classes respecting the Bayes boundaries (intersection of probability densities, to minimize the number of misclassifications).

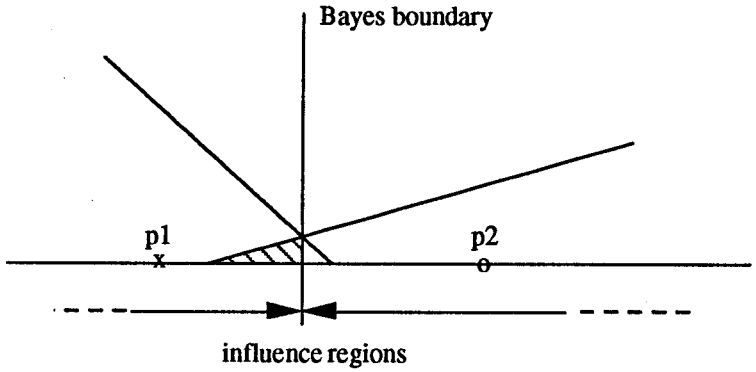

Fig. 2. LVQ1 with optimal decision boundaries

The hatched area on the left of the Bayes decision line represents the probability density of the stimuli from class B, which influence the position of prototype p1 according to equation (3). Since the probability density of class A is greater than the probability density of class B in the whole influence region of prototype p1, it may be approximated that the influence of the B-class stimuli represented by the hatched area is canceled by an identical density area of A-class prototypes (dotted area in figure 3), the prototype moves determined by equations (2) and (3) having identical amplitudes and opposite signs. Let us mention that this is only an approximation, since the compensation of moves induced by equations (2) and (3) depends on the order of presentation of stimuli and of the initial location of prototypes.

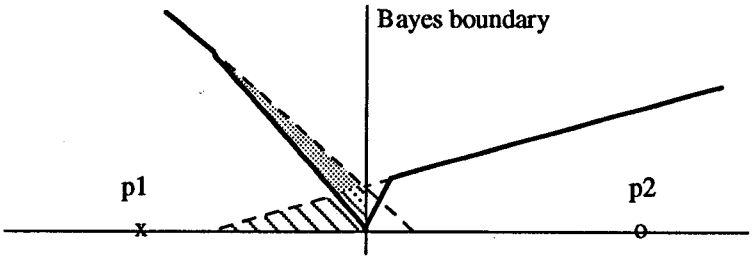

Fig. 3. Approximated equivalent distribution

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 209-214

The plain line in figure 3 then represents an equivalent probability density for the LVQ1; with the above-mentioned assumptions, the resulting distribution of prototypes will be similar.

3.2. Estimation of optimal separation

We assume that the probability densities we have to estimate are represented by the plain line in figure 3, according to the hypotheses of section 3.1. The challenge is now to define how to set the decision boundary line between prototypes p1 and p2. Nearest-neighbour criterion of equation (1) is not optimum; a better criterion will take into consideration the slope of the two probability densities near the boundary, and divide the [p1,p2] segment in an appropriate way.

To estimate where the segment [p1,p2] must be divided, two hypotheses must be done :

- the probability density must be linear into the influence regions of prototypes p1 and p2 (i.e. a linear approximation of the initial distribution must be possible, and the number of prototypes must be sufficient to avoid that the influence regions of p1 and p2 extend to the breakpoints of the probability densities)
- the proportion of prototypes between classes must comply with their a priori probabilities (equally probable classes are thus not required).

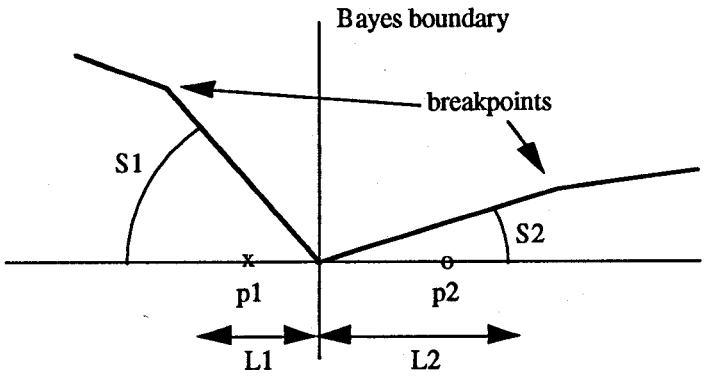

Fig. 4. Estimation of optimal separation

Let us define now (see figure 4) :

- S1 and S2 respectively the slopes of A and B probability densities near the boundary
- L1 and L2 respectively the dimension of the influence regions of prototypes p1 and p2

The second above-mentioned hypothesis assumes the same probabilities to find a stimulus in the influence regions of p1 and p2. The surfaces under the probability densities lines are thus equal :

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 209-214

$$\frac{S1.L1^2}{2} = \frac{S2.L2^2}{2} \tag{4}$$

According the equilibrium point of equation (2), prototype p2 will be situated at a distance C from the origin given by:

$$\int_0^C x.(C-x) = \int_C^{L2} x.(x-C) \tag{5}$$

which gives C=(2/3).L2. The average of the distances between stimuli x and p1 can now be estimated by:

$$\int_0^C \left(\frac{2}{3}.L2-x\right).S2.x\ dx + \int_C^{L2} \left(x-\frac{2}{3}.L2\right).S2.x\ dx = \frac{8}{81}.S2.L2^3 \tag{6}$$

This equation is of course also valid for class A, replacing S2 and L2 respectively by S1 and L1. We can then compute the ratio of these distances averages, using equation (4) :

$$R = \frac{S1.L1^3}{S2.L2^3} = \frac{L1}{L2} \tag{7}$$

On the other hand, the ratio R can easily be repeatedly computed during the adaptation of the prototypes location following equations (2) and (3). Each prototype memorizes a supplementary information D(a), besides its coordinates and its class, which is repeatedly adapted each time equation (2) is used according to:

$$C(a) = C(a) + \alpha\ |x(j)-p(a)| \tag{8}$$

After a sufficient number of iterations, C(a) will represent the distance average between prototype a and all stimuli falling in its influence region. The ratio of parameters C for two prototypes gives thus L1/L2, according to equation (7). By this way, the distances from p1 and p2 where the boundary is to be fixed are known (these distances are proportional to L1 and L2, according to the result of equation (5)).

3.3. Estimation of prototype density in each class

In the previous sections, it was mentioned that the number of prototypes in each class must be proportional to their a priori probabilities. To achieve this goal, the P prototypes are generally chosen as the first P samples of the distribution (the first P stimuli), assuming a good representation of this set. This is however not always true, especially if P is small; in this case, adaptation of the class of some patterns must be achieved.

A simple and relatively good way to adapt the class of the prototypes is illustrated here for a two-class distribution (classes A and B), and can easily be extended to more classes.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 209-214

Let us assign to each prototype a two-bits class vector v(a), whose initial value is (0,0). Let us have a 'global class vector' V also initially set to (0,0). Each time a stimulus is presented, the nearest prototype must be determined to allow adaptation according to equations (2) and (3). Having determined this nearest prototype, we take this opportunity to adapt its class vector by adding an increment of (1,0) if the stimulus belongs to class A, and (0,1) if the stimulus belongs to class B. The global class vector V is also adapted in the same way. The vector V continuously represents thus the proportion of stimuli from both classes. Each K iterations (K being a parameter to be adjusted depending on the size of the stimuli set), the proportions estimated by V are compared with those of the actual prototypes. If there is a difference, let us say that the number of prototypes in class B is too big with regards to class A, one prototype of class B must change its class and be assigned to class A. This prototype is determined among the prototypes in class B by taking the one whose class vector v(a), after normalization to 1, is the most different from (0,1); this prototype is indeed the one which was the most often influenced by prototypes of class A, and thus the closest from the boundary.

This principle can be extended to more classes, by choosing the dimension of the class vectors equal to the number of classes, and by adding vectors like (0, 0, 0, 1, 0) at each presentation of a stimulus.

4. Conclusion

The enhancements described in this paper were applied to 1-D and 2-D pattern classifications, with overlapping distributions. The performances are much closer from the optimal Bayes classification than with the standard LVQ1 procedure, in terms of number of correct classifications. Simulations were made on distributions where the Bayes limit for correct classifications is about 95%, with about 20% of the stimuli in overlapping regions. The average of our simulations gave approximately 13% of misclassifications with the LVQ1 algorithm, and 8% with our suggestions. The 'optimal separation' derived in section 3.2 is of course only valid for linear approximations of the probability densities; this approximation is however generally verified when the number of prototypes is sufficient. Further simulations will evaluate the proposed enhancements of the LVQ1 algorithm on real multi-dimensional and multi-class databases.

References

1. T. Kohonen: Self-organization and associative memory, 2nd edition, Springer-Verlag, Berlin, 1988.
2. C. Jutten, A. Guerin, H.L. Nguyen Thi,: Adaptive optimization of neural algorithms, in: A. Prieto ed., Artificial Neural Networks, Springer-Verlag Lecture Notes in Computer Sciences n°540, Berlin, 1991.

Acknowledgments

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