ELSEVIER

Neurocomputing 35 (2000) 3–26

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Class separability estimation and incremental learning using boundary methods

José-Luis Sancho[a,*,1], William E. Pierson[b], Batu Ulug[b,2], Aníbal R. Figueiras-Vidal[a,1], Stanley C. Ahalt[b,2]

[a]*ATSC-DI, Escuela Politécnica Superior. Universidad Carlos III Leganés-Madrid, Spain*
[b]*Department of Electrical Engineering, The Ohio State University Columbus, OH 43210, USA*

Received 7 January 1999; revised 5 April 1999; accepted 10 April 2000

Abstract

In this paper we discuss the use of boundary methods (BMs) for distribution analysis. We view these methods as tools which can be used to extract useful information from sample distributions. We believe that the information thus extracted has utility for a number of applications, but in particular we discuss the use of BMs as a mechanism for class separability estimation and as an aid to constructing robust and efficient neural networks (NNs) to solve classification problems. In the first case, BMs can establish the utility of a data set for classification. We demonstrate experimentally that the derived ranking is consistent with alternative ranking techniques based on Bayes error (ε). Finally, BMs are used as sample selection (SS) mechanism to train NN by means of gradient algorithms. In particular, elliptic BMs (EBMs) are used to select samples so that the initial partial training set is linearly separable. In a progressive way, new samples are added to the training set solving the problem in an incremental manner. Multi-layer perceptrons (MLPs) and radial basis functions (RBFs) have been used in this work. Our results show that the probability of being trapped in a local minimum is clearly reduced when EBMs are used, making the training independent of the initial weight values. Also, the effect of the very noisy samples and outliers is reduced when the SS-EBM algorithm is employed, so we propose this method as a robust procedure to train NNs by means of a gradient learning rule.

Keywords: Classification neural systems; Class separability estimation; Bayes' error estimation; Gradient algorithms; Sample selection strategies

* Corresponding author. Tel.: + 34-91-6249173; fax: + 34-91-62-49430.
E-mail address: jlsancho@tsc.uc3m.es (J.-L. Sancho).

[1] José-Luis Sancho and Aníbal R. Figueiras-Vidal were partially supported by grant from CICYT Project TIC96-0500-C10-03.

[2] Batu Ulug and Stan Ahalt were partially supported by grants from the Air Force Office of Scientific Research and the U.S. Army Research Lab Programming Environment and Training program.

0925-2312/00/$ - see front matter
PII: S0925-2312(00)00293-9

1. Introduction

The performance of a classification system is dependent upon the data presented to the system. If the data provided is not sufficiently separable, then the system performance will be insufficient, regardless of the classifier used. This concept is captured by the notion of Bayes' error. From a statistical pattern recognition (SPR) framework, the Bayes' error ε, is the minimal error rate obtainable if the statistics of the data are known and is independent of classifier. Thus if the Bayes' error of the data is larger than the acceptable error rate for the classification system, there is no hope of the classification system achieving its design goals. The consequence is that new data must be obtained which may help reduce the Bayes' error to acceptable levels.

Unfortunately, estimation of Bayes' error is a daunting task. Not only does Bayes' error estimation require probability density function estimation, which is known to be an ill-posed problem, but it also suffers from the curse of dimensionality, and normally requires numeric integration in high-dimensional spaces. Thus, ε is either impossible to obtain or difficult to estimate for all but relatively simple distributions [9,25,17].

The proliferation of different class separability measures, as those described in [10,19,7,1], is evidence of the difficulties associated with estimating ε. The large number of alternative class separability measures also suggests that none of techniques provides a universally successful measure. Each alternative measure of class separability has strengths and weaknesses. See [28] or [6] for more detail on measures of class separability.

Once it has been determined that a given data set representing multiple classes is sufficiently separable, a classifier then needs to be designed whose performance is comparable to the predicted performance. Some of the most popular classifiers are based on NNs because of their proven capabilities to solve complex pattern recognition problems. Among the different types of NNs, the multi-layer perceptron (MLP) and the radial basis function (RBF) are among the more popular and frequently used architectures; the typical structures of MLP and RBF networks are shown in Figs. 1 and 2, respectively. The MLP neural network is normally trained with the standard back-propagation (BP) algorithm which modifies the network weights according to a gradient descent learning rule [23]. RBF neural networks can be trained according to several strategies, one of which is using a supervised selection of the RBF parameters by means of a gradient algorithm [15]. The main drawback of many of these gradient descent procedures is slow and/or unreliable convergence. The major reasons for convergence difficulties are the existence of regions where the error function is very flat, and saddle points and local minima where the network weights can be trapped. Thus, several factors affect learning in NNs, including the type of the error function to be optimized, the network architecture, the learning algorithm, the values of the training parameters (specially the learning rate), the initialization of the network weights, and the strategy used to select the samples [16,12].

In this paper, we discuss the use of boundary methods (BMs) for statistical pattern classification. The use of BMs for class separability estimation is described in

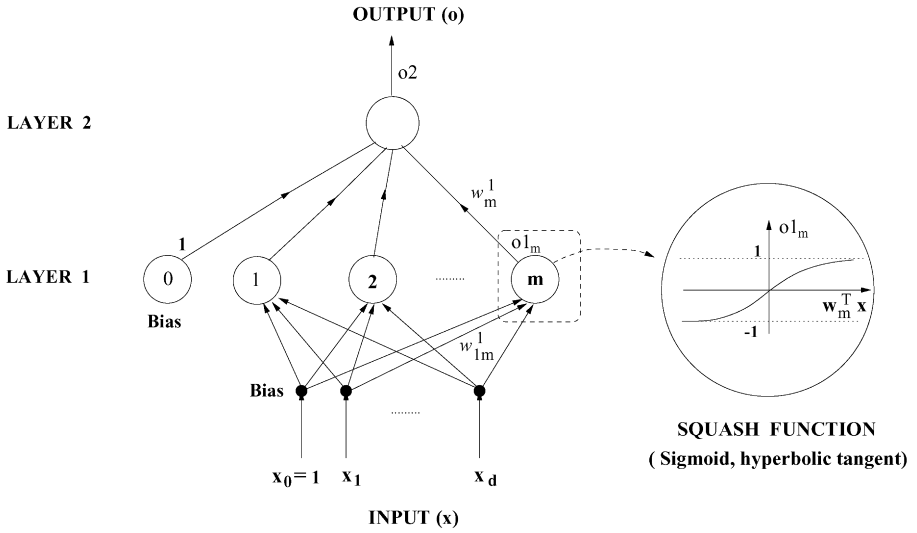

Fig. 1. A MLP neural network with one hidden layer and one output node. Normally, squash functions (logistic sigmoid or hyperbolic tangent) are used as non linear node functions.

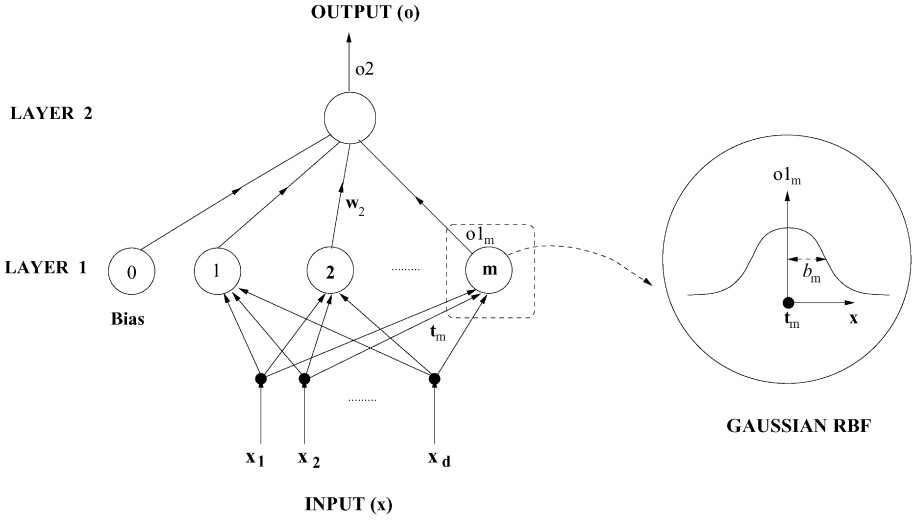

Fig. 2. RBF architecture with spherical Gaussian radial basis functions. t_i are the Gaussian's centroids, b_i are the Gaussian's spreads, and $\boldsymbol{w}$ are the weights of the output layer. Output nodes use to have linear or squash functions (logistic sigmoid or hyperbolic tangent).

Section 2.1. In Section 2.2, BMs are used as a sample selection (SS) mechanism to train neural networks in an efficient way. Several experimental results are shown in Section 3. We conclude with a summary of our results.

2. Boundary methods

2.1. Boundary methods for class separability estimation

The motivation for the development of BMs can be described conceptually as follows. Misclassified samples can be thought of as those which lie on the wrong side of a given decision boundary (see Fig. 3 for an illustration of these concepts). If the decision boundary is the Bayes boundary, the error asymptotically becomes the Bayes error as the number of samples approaches infinity. By surrounding groups of samples of each class with a boundary, class overlap regions are formed. Since most classification systems derive decision boundaries located in the overlap regions between classes, a count of the samples in this region provides a measure of the classification error. Collapsing these boundaries in a structured manner and counting the samples in the resulting overlap regions provides a series of decreasing values all related to the classification error [21,20]. This assumption is more valid if the method used to shrink the boundaries continually results in an overlap region that contains the decision boundary.

This is the essence of BMs, counting the samples found in overlap regions formed by boundaries. Granted, for this general notion, terms such as "boundary", "measure", "structured" and "shrunk" are not precisely defined. These terms will become clearer as the discussion progresses.

The core of the BM concept is determining the trajectory (that is, the sequence of recorded counts) of samples found in one or more overlap regions [24]. Thus, a description of this procedure is given first. The steps for determining the trajectory between two classes are:

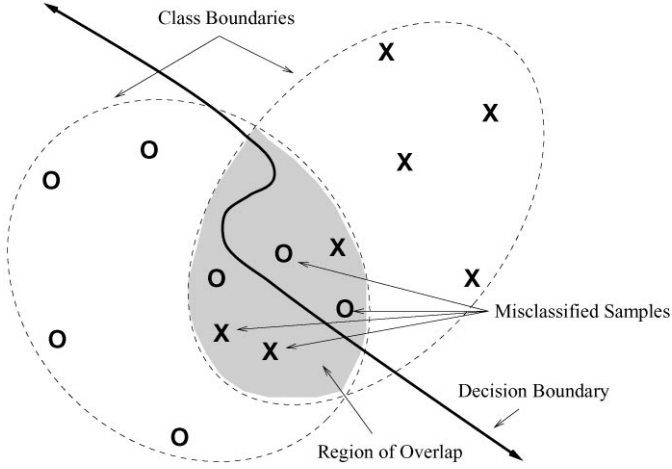

Fig. 3. An illustration demonstrating the concepts of class separability, regions of overlap, and misclassification.

- Form class boundaries.
- Find the number of samples in overlap region.
- Reduce, or collapse the class boundaries.
- Continue this process until the overlap region contains zero samples.

This process is demonstrated in Fig. 4. The figure shows the formulation of the boundaries, the samples in the overlap region, the zero volume overlap (ZVO) point, and the resulting overlap count (OC) as a function of the iteration. ZVO refers to the point where the intersection of the decayed boundaries contains zero samples. OC is the number of samples in the overlap region as a function of iteration through the algorithm. Here ellipsoidal boundaries are used with the corresponding estimated mean and covariances representing the constant-density contours of the classes.

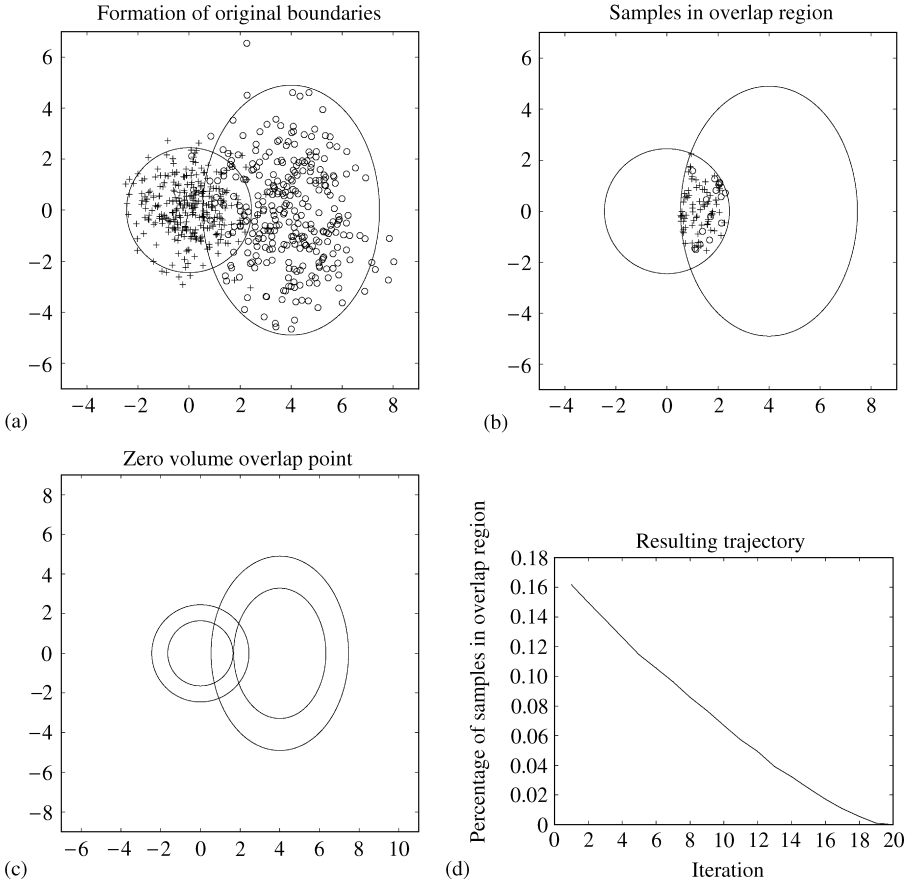

Fig. 4. The concept behind the BM process: (a) The initial data and boundaries after outlier removal, (b) the samples in the overlap region (c) the beginning (outer) boundaries and the boundaries located at the ZVO point (inner), (d) the resulting percentage of samples enclosed by the overlap region as the boundaries step down from the initial to the ZVO boundaries.

There are three reasons for the selection of ellipsoidal boundaries. First, due to their mathematical form, the number of samples enclosed in the overlap region is easily determined. Second, it is often appropriate to use a Gaussian assumption [11]. Third, we can use a collection of ellipsoidal boundaries to represent multi-modal data. While ellipsoidal boundaries are used for our examples, any justifiable boundary can be used in general, thereby relaxing any normality assumption. The use of ellipsoidal boundaries is a natural choice for any distribution that is elliptically contoured and extends easily for elliptically contoured mixture densities [27]. An alternative method to using mean and covariance estimates to form the ellipses is to use minimum volume bounding ellipses [26]. This divorces the procedure from any assumed distribution, and allows for the formation of tight ellipse boundaries in those situations where the form of the distribution is known but insufficient samples exist for an accurate covariance estimate.

As noted above, alternative boundaries can be used and Fig. 5 demonstrates some of these alternatives.

There are several possible methods that can be used for reducing the boundaries. One intuitive method is to decay the boundaries such that each mode loses samples, i.e., samples fall outside of the boundary, at a linear rate. Here, mode refers to the modes of a distribution. This approach is used for the experiments described in this paper. Another method is to reduce the boundaries so the a posterior probability levels are equal. This is useful for collapsing to a point close to the Bayes decision

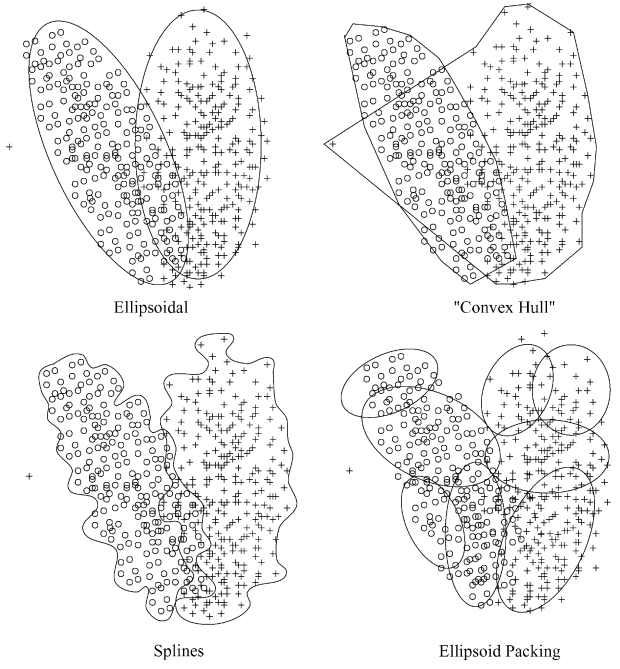

Fig. 5. Different possible boundaries that may be useful in boundary methods.

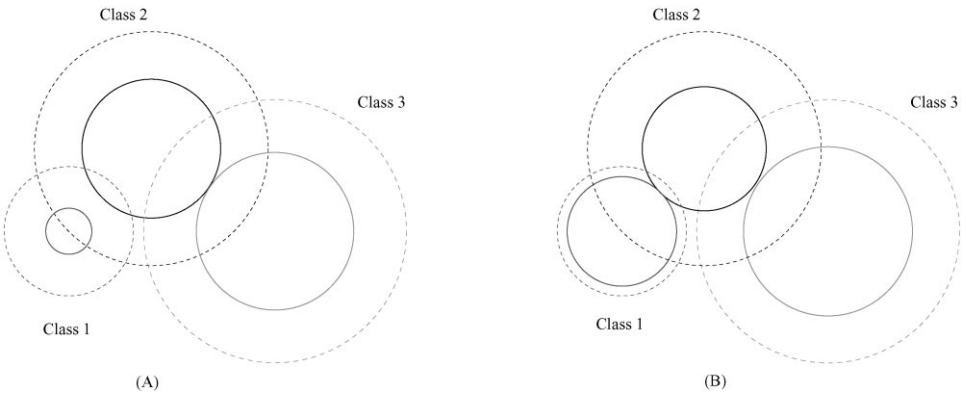

Fig. 6. An illustration of the resulting ZVO points obtained by (A) simultaneously shrinking all boundaries and (B) shrinking boundaries iteratively. Dashed lines are the original boundaries. Solid lines are the ZVO boundaries.

boundary. There also exists an analytic method for finding the tangent point of two ellipses with equal area [8].

Regardless of the method used, the boundaries are decayed until zero samples occur in the overlap region. We call this the zero volume overlap (ZVO) point. The ZVO tells us the point at which the distribution's confidence intervals are such that all points enclosed by the resulting boundaries can be correctly classified with a piecewise linear classifier.

This procedure forms the kernel of the BM technique and works for the unimodal, two-class case. In order to extend the technique to the multimodal, multi-class case, modifications are needed for the reduction of boundaries because, if all boundaries are reduced by equivalent amounts, the resulting ZVO point can be such that only the boundaries between two modes of different classes are tangential, as can be seen in Fig. 6. All other boundaries may have been reduced to such a degree that they are well separated.

Thus an alternative procedure was developed for reducing the boundaries for the multimodal, multi-class case. Here we adopt an iterative method which modifies the boundaries at each step for only a pair of modes. The modes to be shrunk must meet two criteria. First, each mode must belong to a different class. Second, the number of samples in the pair's overlap region must be more than the number of samples in any other overlap region that passes the first criteria. The resulting ZVO is such that all boundaries are close.

A normalization step is needed to compensate for the iterative nature of this method for pairwise shrinking. The procedure is normalized by starting with the initial boundaries and shrinking to the ZVO point as found before, but each mode boundary is shrunk at each step so that the ZVO is reached in a fixed number of steps. Thus, an OC is formed where the number of steps taken is fixed. This normalization assures the sum of the trajectory values, called the overlap sum (OS), is not artificially

inflated due to the iterative stepping procedure and allows OCs obtained from different FE procedures to be compared fairly.

For the BMs described up to this point, we require data labeled with both class and mode. Mode information is needed because the procedure does not have any mechanism for determining the modes (clusters) for a given class. Several clustering routines exist that can estimate the model order of each class. Furthermore, as discussed below, we can actually use BMs to estimate the modes – thus permitting us to directly process labeled data.

For each class, a mixture of Gaussian's assumption was used [11]. Based on this assumption, we can use a maximum likelihood estimation approach for clustering the data, assuming the number of modes of each class is known. Starting with a model order of one, OC trajectories for each mode are found using the routine outlined above. However, since no overlap region exists, the routine was modified to count the number of samples enclosed within the one bounding ellipsoid. Since the assumption was a mixture of Gaussian's and we are decaying the boundary such that the number of samples enclosed decays linearly, we know the projected rate of decay. If the actual rate of decay varies significantly from the linear rate, that region (boundary) was flagged as requiring another mode.

To estimate the location of the mean of the new mode along the flagged boundary, the mean of all samples within a fixed distance of this ellipsoid was found. The direction of the mean points to the location for the estimate of the new mode mean. We then repeat the maximum likelihood clustering technique with an increased model order.

With these modifications the BM procedure performs class separability estimation for the multimodal, multi-class case. Fig. 7 presents a flowchart of this procedure.

We reiterate that the plot of the trajectory indicates how the different-class distributions are enmeshed. The intuition as to why this measure provides information related to ε is as follows. As the classes become more separable, the ε decreases. In order for this to occur, the area under the non-maximal a posteriori probabilities must be small. Since the OS constitutes some measure of this information, it is related to ε. Thus, as the OS decreases, so does ε and therefore class separability increases.

2.2. Boundary methods as SS procedure

When NNs, such as MLPs and RBFs, are trained by means of a standard gradient algorithm, e.g. BP, a random presentation of the samples is generally used. In this case, a teacher informs the network of whether the decision is correct or not, but is not concerned about how to present the learning samples. Moreover, the training set usually contains samples which are either redundant or do not provide equally useful classification information. For this reason, different approaches have been developed to examine the effect of sample presentation and selection on learning [29,18,22,5,30]. For example, MacKay [18] and Plutowski and White [22] propose a similar approach to find critical samples for the training process. Both methods require the calculation of the second derivatives (Hessian matrix) so they can be computationally expensive. Zhang [30] describes a method for selecting critical examples which does

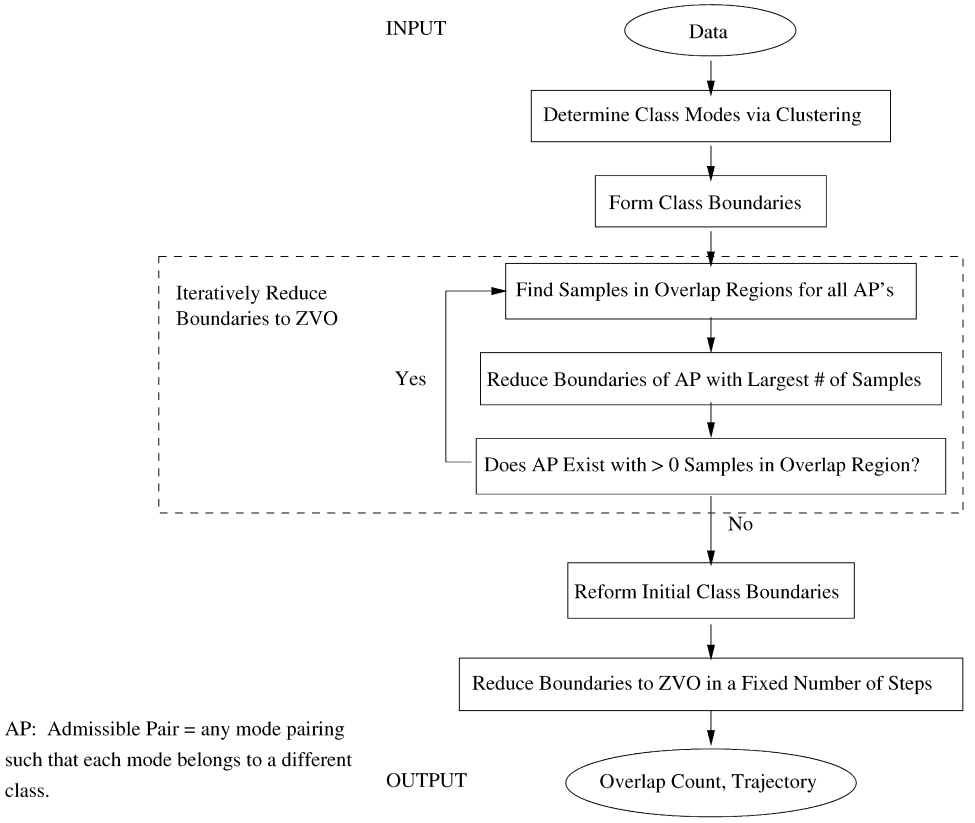

Fig. 7. Flowchart for the BM procedure for class separability estimation.

not require the second derivatives. Although Zhang's method can significantly improve the speed and generalization performance, it has some limitations. For example, Zhang's method can become stuck in a local minimum during training, and it's performance on very noisy data can be poor. This is due to the fact that the method selects as critical samples those samples with maximum error function values, which can be problematic when samples with significant numbers of outliers.

Using Zhang's terminology, these previously mentioned methods can be considered as *incremental learning* (IL) procedures. Instead of training the network on the entire training set, $D_N = \{(\boldsymbol{x}_1,\boldsymbol{t}_1),(\boldsymbol{x}_2,\boldsymbol{t}_2),\ldots,(\boldsymbol{x}_N,\boldsymbol{t}_N)\}$, where $\boldsymbol{x}_j$ and $\boldsymbol{t}_j$ are the samples and targets, respectively, IL start the learning with a small subset D_{N_1}, $N_1 \ll N$, increasing the training set incrementally. In each step, the training set is increased according to some criterion until the step M where $D_{N_M} = D_N$. Thus, the training task is solved in a progressive way, from the initial situation where the training set is small, until global convergence is reached. Here the term global is meant to refer to the entire training set. An example of incremental classification is shown in Fig. 8. Fig. 8(a) and (a') show

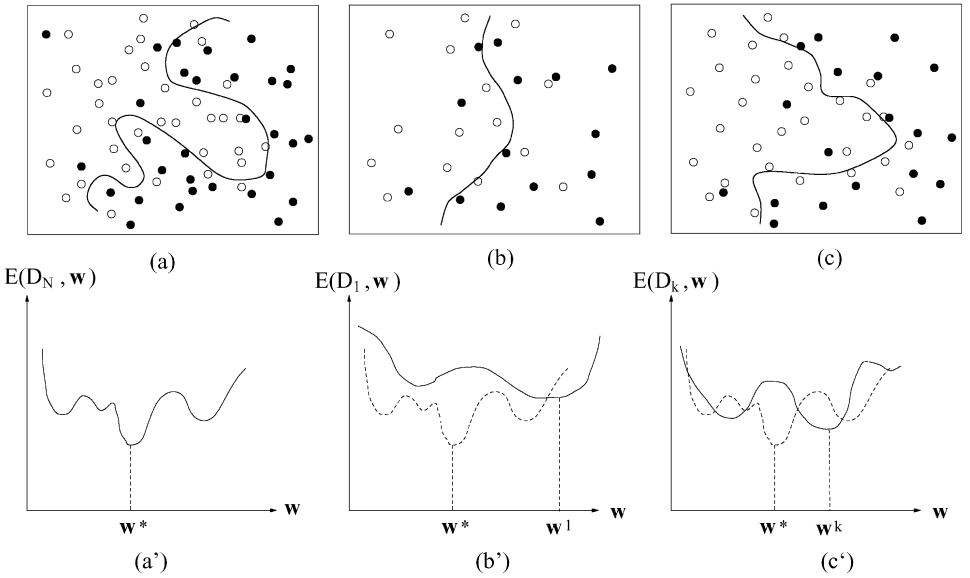

Fig. 8. Example of Incremental Learning (IL). (a) and (a') show the classification problem to be solved and the global (all training samples) error surface, respectively. The optimum solution is a point in the weight space at the minimum of this surface, $\boldsymbol{w}^*$. Two steps of the IL (the initial and the kth steps) are shown in (b), (b') and (c), (c'), respectively. We can see how the original problem (a) is solved in a progressive way.

the classification problem and the global error surface, respectively. The optimum solution will be obtained with a set of weights around the minimum of this surface, $\boldsymbol{w}^*$. Two steps of the IL are shown in Fig. 8(b), (b') and (c), (c'), respectively.

Here, we propose a new incremental learning procedure based on BM which is not based on a particular error function. In our method, BM are used to incrementally select the samples (SS) of the training set in such way that several important advantages are obtained. The main idea is quite simple and will be described considering elliptic boundaries (EBM).

2.2.1. SS-EBM method

Consider a two-class classification problem. First, we enclose most of the samples (95%, for example) of each class by means of ellipses (hyper-ellipsoids, in a general case) estimated from the data. Second, the ellipses are shrunk until they are tangent (ZVO) and, subsequently, expanded in a progressive fashion from this *initial* tangent configuration. At each step, the training proceeds using only the data of each class inside the corresponding ellipse. Note that the ellipses at any step are the expansions of the ellipses realized when the previous step reaches partial convergence (corresponding to the error measured over the enclosed samples), defined as occurring when the error becomes stabilized. Training ends when the global convergence (corresponding to the error measured with all the available patterns) is reached.

Suppose that the two-class classification problem is described by training set $D_N = \{(\boldsymbol{x}_1,\boldsymbol{t}_1),(\boldsymbol{x}_2,\boldsymbol{t}_2),\ldots,(\boldsymbol{x}_N,\boldsymbol{t}_N)\}$, which is used to train a neural network using a gradient learning rule. The weights of the network are denoted by $\boldsymbol{w}$; $E(D_N,\boldsymbol{w})$ is the global error function to be minimized (we assume that it is a derivable function of $\boldsymbol{w}$), and $E_{\min} = E_{\min}(D_N,\boldsymbol{w})$ is the global error goal. Then, the algorithm can be described as follows.

Algorithm:

(1) Estimate the mean vectors, $\boldsymbol{m}_j$, and the covariance matrix, Σ_j, from the patterns of each class, C_j, $j = 1,2$. In this manner, a family of ellipses described by a quadratic form, $H_j(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{m}_j)^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{m}_j)$, $j = 1,2$, can be associated with each class.

(2) Calculate the parameters $R_j(M)$ such that

$$Pr\{H_j(\boldsymbol{x}) < R_j(M)\} = 0.95 \tag{1}$$

for $\boldsymbol{x} \in C_j$, i.e., calculate the ellipses $H_j(\boldsymbol{x}) = R_j(M)$ that enclose the most (95%) of the samples.

(3) Shrink these ellipses reducing $R_j(M)$ progressively until the tangent situation (ZVO) is detected, i.e., until the pattern intersection set between the two ellipses is empty. In this situation, the new ellipses are determined by parameters, $R_j(1)$, described by equations $H_j(\boldsymbol{x}) = R_j(1)$, $j = 1,2$, and satisfy

$$\{\boldsymbol{x} \in D_N,\ H_j(\boldsymbol{x}) < R_j(1), \text{ for } j = 1,2\} = \emptyset. \tag{2}$$

Next, the values of $R_j(k)$, $k = 1,2,\ldots,M$, are recorded, and k is set to one ($k = 1$).

(4) Training the network by means of a gradient learning rule using the following training set

$$D_{N_k} = \{\boldsymbol{x} \in C_1,\ H_1(\boldsymbol{x}) < R_1(k)\} \cup \{\boldsymbol{x} \in C_2,\ H_2(\boldsymbol{x}) < R_2(k)\} \tag{3}$$

until satisfying the stopping criterion: $E(D_{N_k},\boldsymbol{w}) < e_{\min}$, or $E(D_{N_k},\boldsymbol{w})$ becomes stabilized. Here, $e_{\min}$ is a partial error goal verifying $e_{\min} \leqslant E_{\min}$.

(5) If $E(D_N,\boldsymbol{w}) < E_{\min}$ or $k > M$, then stopping the training; otherwise, $k = k + 1$ and back to 4.

This incremental learning method presents several interesting properties. First, the SS procedure is not constrained to a particular error criterion and can thus be used with an arbitrary objective function. Second, the initial training set (D_{N_1}) is linearly separable, as Fig. 9(b) shows. Thus, the algorithm becomes very insensitive (i.e., robust) to the weight initialization process because $E(D_{N_1},\boldsymbol{w})$ is very simple and the convergence can be easily achieved. In particular, it can be proved that for an MLP with sigmoidal nodes, SSE criterion function, and linearly separable patterns, the convergence to optimal solutions (i.e., global minimum) is ensured by using standard batch mode BP algorithm [13,14]. Similar result can be stablished for RBF networks with a hidden layer of radial basis function units and sigmoidal output nodes; in this case, the requirement that all the patterns of the training set are linearly separable is replaced by a requirement that all the patterns are separable by hyperspheres or by

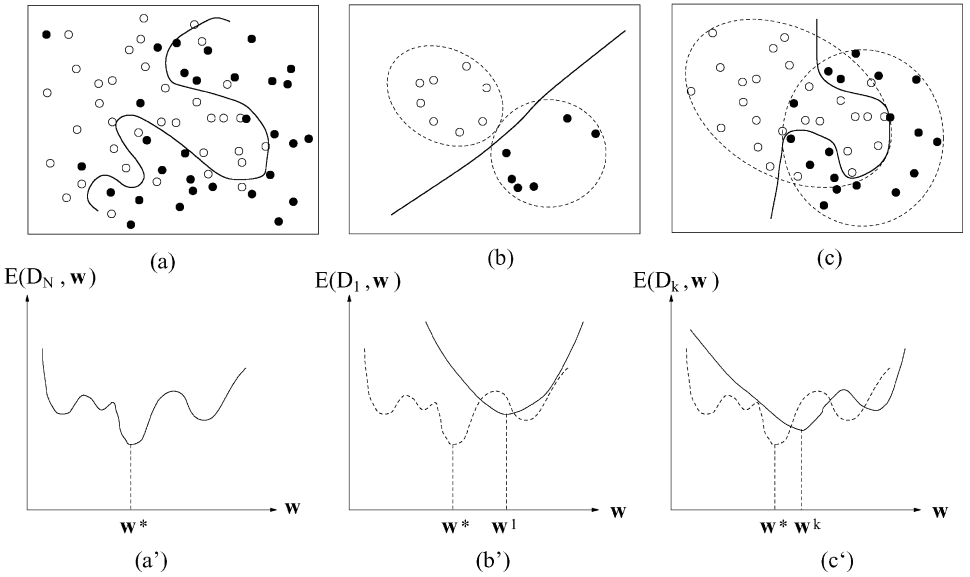

Fig. 9. Example of SS-EBM incremental learning using a multilayered neural network classifier for the same example shown in Fig. 8. Here, the initial error surface shown in (b') corresponds to the tangent situation (ZVO) shown in (b), i.e., to the initial training patterns enclosed by the tangent ellipses. This surface $E(D_{N_1}, \boldsymbol{w})$ is now very simple because the (initial) training set is linearly separable (see (b)). In particular, it can be proved that, for MLP and RBF networks, this initial error function $E(D_{N_1}, \boldsymbol{w})$ is free of local minima. Thus, the convergence of a gradient descent algorithm is guaranteed independently of the initial value of $\boldsymbol{w}$. The incremental learning is controlled by the sequential growing of the ellipses such that the risk of remaining stuck in a local minimum is drastically reduced. Moreover, the effect of the outliers is also reduced because they are not considered in the training set until the last steps of the growing ellipses process, where we can expect the algorithm has already converged.

the more general quadratic surfaces, hyperellipsoids [3,2,14]. This idea is shown in Fig. 9(b) and (b'). Third, starting from a good initial estimate and using incremental learning controlled by the sequential growing of the ellipses drastically reduces the risks of remaining stuck in a local minimum. This property will be experimentally confirmed in Section 3. Finally, the effect of the outliers (and very noisy patterns) is also reduced because they are not included in the training set until the final steps of the algorithm. Fig. 9 demonstrates how outliers are incorporated into the processing in the final steps of the algorithm. In Fig. 9(a), the black sample at upper left corner is clearly an outlier. The intermediate steps of the SS-EBM procedure, as shown in Fig. 9(b) and (c), do not include this sample in the training set because this datum lies outside of the intermediate ellipses. Indeed, the sample is only included in training set when it is finally enclosed by the final (largest) ellipses.

The behavior of the SS-EBM algorithm can be analyzed using a technique similar to that presented by Zhang [30].

Normally, the error criterion function takes the form of a sum of over patterns of an error term for each pattern separately [4], i.e.,

$$E(D_N, \boldsymbol{w}) = \sum_{j=1}^{N} E_j(\boldsymbol{w}), \tag{4}$$

where j indicates the index of examples, and $\boldsymbol{w}$ contains the weights and parameters of the network. Thus, instead of minimizing the global error function (4) directly, SS-EBM tries to minimize a set of error functions $E(D_{N_1},\boldsymbol{w})$, $E(D_{N_2},\boldsymbol{w}), \ldots, E(D_{N_M},\boldsymbol{w})$, where M is the maximum number of steps of the process and sets D_{N_j} satisfy

$$D_{N_1} \subset D_{N_2} \subset \cdots \subset D_{N_M} \simeq D_N. \tag{5}$$

Equivalently, SS-EBM tries to minimize the error function

$$\begin{aligned} E_{\text{SS-EBM}}(D_N,\boldsymbol{w}) &= \sum_{j=1}^{M} E(D_{N_j},\boldsymbol{w}) \\ &= E(D_{N_1},\boldsymbol{w}) + \cdots + E(D_{N_M},\boldsymbol{w}) \\ &= \sum_{j=1}^{N_1} E_j(\boldsymbol{w}) + \cdots + \sum_{j=1}^{N_M} E_j(\boldsymbol{w}). \end{aligned} \tag{6}$$

We can see that minimizing the partial objective functions $E(D_{N_i},\boldsymbol{w})$, $i = 1, 2, \ldots, M-1$ leads also to the minimization of $E(D_{N_j},\boldsymbol{w})$, $j = i, i+1, \ldots, M$. Therefore, we can easily infer that minimizing $E_{\text{SS-EBM}}(D_N, \boldsymbol{w})$ also produces the minimization of the global error $E(D_N,\boldsymbol{w})$ measured over the entire sample set. In this sense, SS-EBM algorithm realizes the global gradient algorithm, i.e., the gradient learning rule applied over the global error function (measured with all the training patterns).

3. Experimental results

3.1. BM for class separability estimation

We argue that BMs can be used to estimating class separability because of a relationship between the OS and ε. While the theoretic relationship between the two values has been completed for selected distributions [20], here we present empirical evidence to demonstrate proof of concept.

In order to determine whether the relationship between OS and ε is valid for different scenarios, several parameters were identified as having a possible impact upon the results. These parameters are:

- number of classes,
- number of modes per class,
- type of distribution,
- data dimensionality,
- proper model order selection.

Experiments were performed where the above parameters were modified. Results from a representative few will be shown here to demonstrate the performance under different conditions. The scenarios demonstrated are as follows:

- Two dimensional, two class, unimodal, Gaussian distributions.
- Two dimensional, three class, multimodal, Gaussian distributions.
- Two dimensional, three class, multimodal, uniform distributions over elliptical supports.
- Eight dimensional, two class, unimodal, Gaussian distributions.

This set is diverse enough to demonstrate the effects of different operating conditions on the relationship between ε and OS.

The results obtained were found by running the following procedure on the above data sets. First $\hat{\varepsilon}$ and OS were found using 50 Monte Carlo simulations on 5000 data samples. Then the mode means were radially expanded from the overall mean.

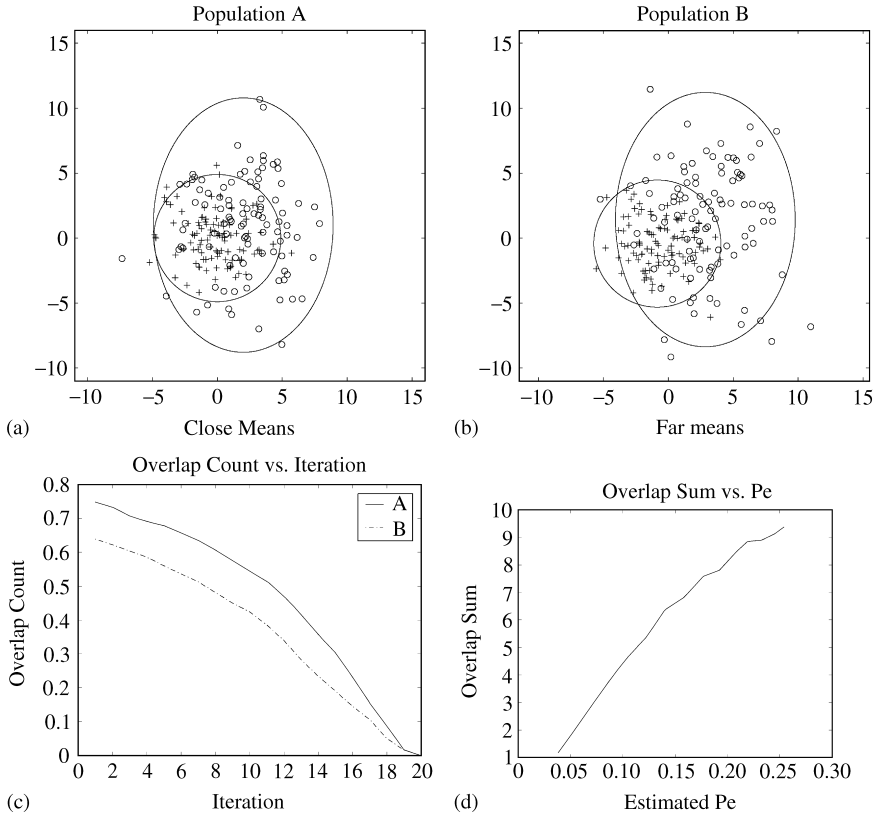

Fig. 10. The 2-class, Gaussian case: (a) original data and boundaries with means closely spaced, (b) original data and boundaries with means spaced further separated, (c) the resulting OC trajectories for cases shown in (a) and (b), (d) the average OS vs. the Bayes error ε, where ε is modified by repeatedly radially separating the mode means.

Dispersing mode means results in a smaller ε value. Again, the OS and $\hat{\varepsilon}$ values were calculated. This process was continued until a sufficiently small $\hat{\varepsilon}$ value was reached. Thus, for each scenario the relationship between $\hat{\varepsilon}$ and OS was obtained.

In Fig. 10 we show simulation results for a two class unimodal Gaussian case in two dimensions. The original (starting) distributions are shown in Fig. 10(a) as represented by their 95% confidence intervals (seen here as ellipsoids) along with a scatter plot of some samples drawn from these two-class distributions. In Fig. 10(b) the same plot is shown after the class means have moved away from each other slightly. The latter case is clearly more separable and thus results in a smaller ε. This is reflected in Fig. 10(c) which shows the class trajectories obtained by the BM algorithm for both of the above mentioned situations, referred to as A and B, respectively. Since case B is more separable than case A, the trajectory curve for B is below that of A. Finally, the OS is plotted against the ε estimate and shown in Fig. 10(d) displaying a near linear relationship between the two.

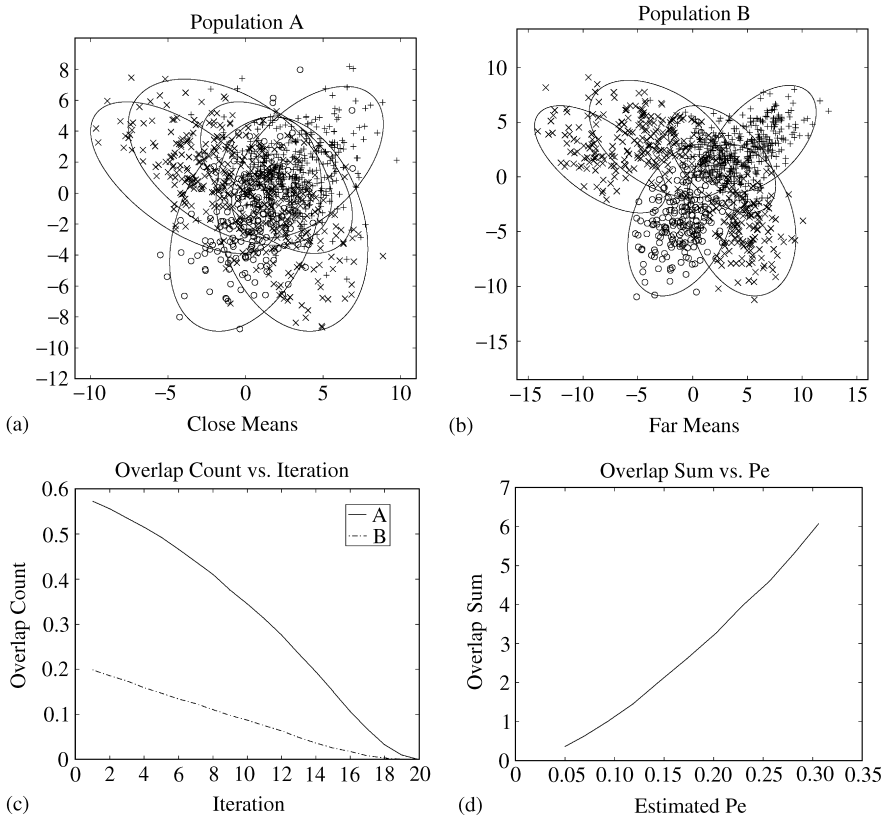

Fig. 11. Multiclass, multimode, Gaussian case: (a) original data and boundaries with means closely spaced, (b) original data and boundaries with means spaced further part, (c) the resulting OC trajectories for cases shown in (a) and (b), (d) the average OS vs. the Bayes error ε, where ε is modified by repeatedly (radially) separating the mode means.

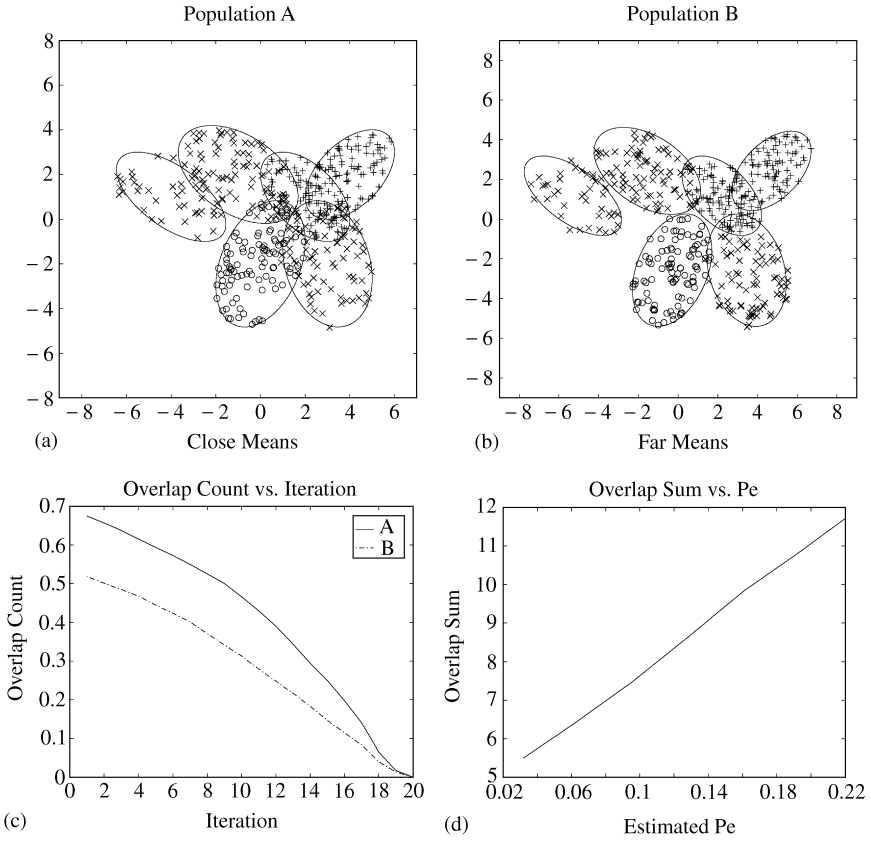

Fig. 12. Multiclass, multimode, uniform over elliptic support case: (a) original data and boundaries with means closely spaced, (b) original data and boundaries with means spaced further part, (c) the resulting OC trajectories for cases shown in (a) and (b), (d) the average OS vs. the Bayes error ε, where ε is modified by repeatedly radially separating the mode means.

Figs. 11 and 12 show the same plots as seen in Fig. 10. However, these show the plots for the multi class, multimodal case in two dimensions. The only difference between these figures is that Fig. 11 is for Gaussian distributions whereas Fig. 12 represents the case where the underlying pdf's are uniform over an elliptic support. Again the figures support the claim that as the probability of error increases, so does the OS.

To show the effects of dimensionality on the relationship between ε and OS we use the eight dimensional, two class, Gaussian data described by Fukunaga [11, Chapter 2] as the I-Λ data set. However, since our procedure varies means in order to find samples along the ε axis, the mean values are not identical to I-Λ data set but the relative vector directions are the same. Fig. 13 shows the resulting OC trajectories for different means as well as the relation between ε and OS.

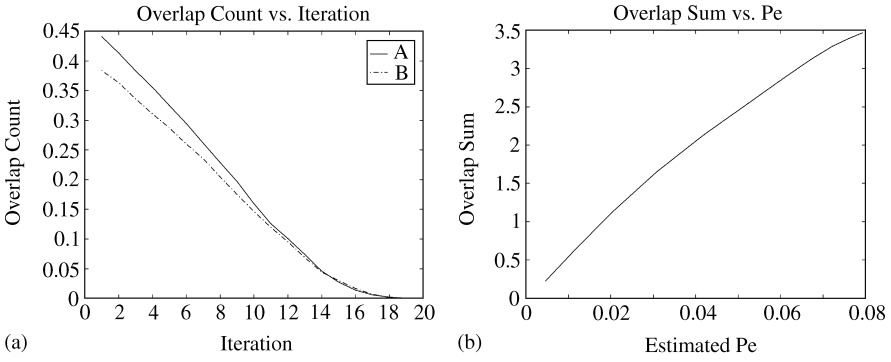

Fig. 13. 2-class, 8-dimensional case: (a) the OC trajectories for the I-Λ dataset for different means, (b) the average OS vs. the Bayes error ε, where ε is modified by repeatedly radially separating the mode means.

3.2. BM as SS

Here we show how BMs may be used to improve the results obtained when a MLP and a RBF are trained by means of a gradient algorithm. In particular, we present classification problems which demonstrate how a gradient algorithm (in batch mode) can be combined with a SS mechanism which is controlled by the Elliptic BM, SS-EBM Method. We compare the decision boundaries obtained when the gradient algorithms are used to train both NNs with and without the use of BM.

3.2.1. BMs used to train a MLP-NN

The problem consists of two classes, both uniformly distributed inside a two dimension rectangular region. Fig. 14(a) shows the training samples of the two classes and the ideal border. Note that we have specifically devised a problem in which a "spur" in the data requires the formation of a relatively complex set of boundaries. The total number of samples in the training set is 600. Another set of 6000 samples is used for testing. Both algorithms use the same architecture (3 hidden nodes), training and test sets, small random initial weights, and the same learning rate parameter (0.005).

The error function used in all simulations is the sum-squared error (SSE). Note that this is not the most appropriate criterion to use for a classification problem for two reasons. First, SSE is derived from maximum likelihood under the assumption of Gaussian distributed target data while the target values typically used in classification problems are binary (or discrete) codes and hence far Gaussian [4]. Second, the SSE criterion places emphasis on points where the probability of the data is large, rather than on points near the decision surface, which are more critical in order to define the surface and where the probability of occurrence is normally low [10].

$E_{\min}$ ($SSE_{\min}$) and $e_{\min}$ are all set to zero so that training is stopped when the error function is stabilized.

We performed ten simulations of the SS-EBM procedure, and ten simulations of the BP algorithm. In the first case all available samples form the training set. In the

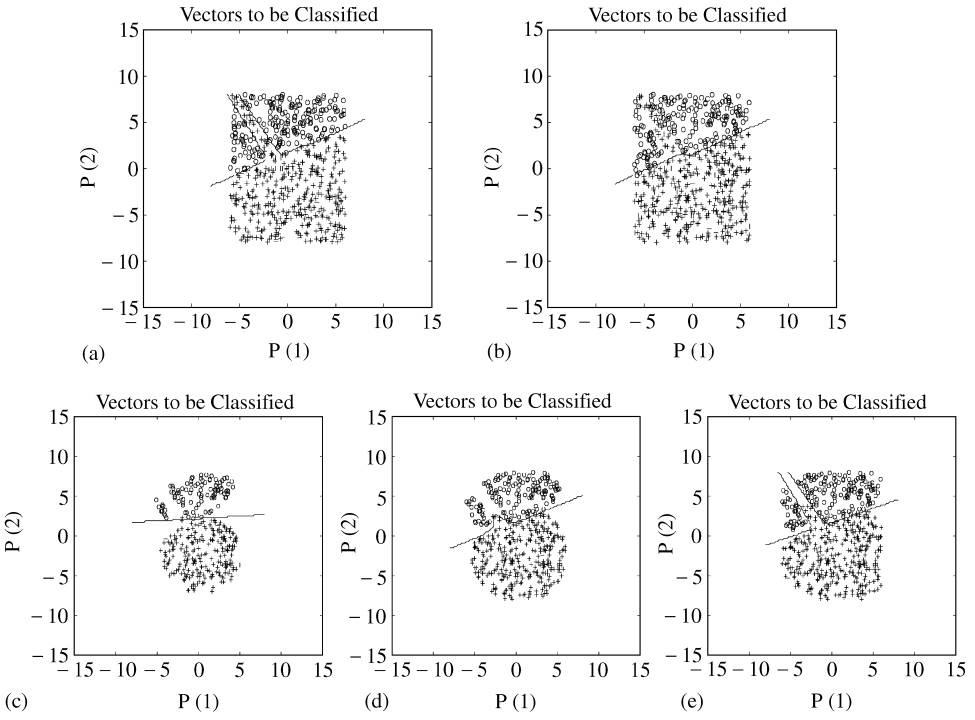

Fig. 14. (a) Problem to be solved, and the optimal decision border; (b) final state obtained by the standard BP algorithm in all the ten simulations, which is clearly a clear local minimum; (c)–(e) evolution of the decision boundary during the training process with the SS-EBM procedure.

second case, each class has an associated time-varying ellipse and only the samples inside the ellipse are included in the training set at step k during the training process. The ellipses are expanded from the initial (ZVO) configuration in which only a subset of high-confidence samples are enclosed, to the final configuration in which the enclosed samples comprise a high percentage (95%) of the total samples. As discussed above, this method of sample selection has two significant benefits. For example, outliers (and very noise patterns) have a significantly diminished impact on training the classification border. Additionally, we solve the problem in a progressive fashion, starting with a problem that is linearly separable, and then gradually inducing a more complex border as additional samples are taken into account – thus reducing the probability of converging to a local minimum and making the convergence inconsequential of the initial values of the network's weights.

Simulation's results with the standard BP algorithm highlight the difficulty of obtaining a good decision boundary. Effectively, standard BP becomes trapped in a local minimum for this example as Fig. 14(b) shows. This phenomenon occurred for all of our 10 simulations, regardless of the initial conditions we selected. Nevertheless, when the SS-EBM procedure and BP learning are combined, convergence was

achieved in 90% of all the test cases. Fig. 14(c)–(d) and (e) shows the sequence of boundaries obtained during training using SS-EBM algorithm.

3.2.2. BMs used to train a RBF-NN

In this case, the problem consists of two classes, both uniformly distributed inside a two dimension rectangular region. Fig. 15(a) shows the training samples of the two classes and the optimal border. This problem was constructed using a two-layer RBF net with 5 hidden nodes (Gaussian RBFs) and one linear output node. The gradient algorithm used in these simulations is a slightly different version of that one shown in [15]. Table 1 describes this version.

For this example we compare the classification border derived using (1) an two-layer RBF net with 4 hidden nodes and one linear output node trained with a gradient algorithm with (2) the same RBF net trained using gradient algorithm, but with samples progressively defined by the SS-EBM Method.

The gradient algorithm used is based on the use of spherical Gaussians defined by

$$G(\boldsymbol{x}, \boldsymbol{c}) = \exp(-b\|\boldsymbol{x} - \boldsymbol{c}\|^2),$$

where $\boldsymbol{x}$ is an input vector, $\boldsymbol{c}$ is the mean, and the parameter b controls the spread. The centroids (4 two-dimensional vectors), second layer weights (4 components vector) and the parameter $\boldsymbol{b}$ (four-component vector) are trained with learning rates of 0.6, 0.3 and 0.001, respectively, and the training is terminated when the mean-square error (MSE) – a variant of the SSE – reaches a relatively constant value (stabilized error).

Repeated simulations show that, without using SS-EMB, the gradient algorithm which employs all of the training samples converges to a good solution only about 10% of the time, typically converging to a local minima with a large MSE. In contrast, when the SS-EBM procedure is employed convergence to a good solution is reached for all simulations with a mean probability of error of about 0.07. Fig. 15(b) shows the decision boundary for a particular simulation when EBM is applied.

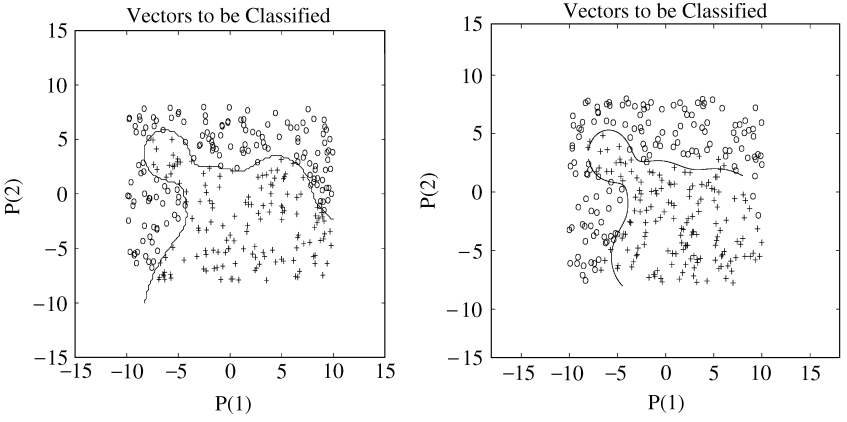

Fig. 15. (a) Problem to be solve an optimal decision border; (b) solution for gradient algorithm controlled by Elliptic BM (SS-EBM Method).

Table 1
Gradient algorithm for train RBF networks

Objective: minimize the mean square error by means of varying the network's parameters in the opposite direction of the gradient

$$E = \frac{1}{2N} \sum_{j=1}^{N} e_j^2$$

$$\gamma(n+1) = \gamma(n) - \mu_\alpha \frac{\partial E(n)}{\partial \gamma(n)}$$

where γ can be w_i,$\boldsymbol{c}_i$, or b_i with $i = 1,2,\ldots,H$ (number of hidden nodes), N is the number of training samples and $\alpha = 1,2,3$ consistent with:

(1) *Linear weights* (output layer)

$$\frac{\partial E(n)}{\partial w_i(n)} = \frac{1}{N} \sum_{j=1}^{N} e_j(n) o1_{ij}(n)$$

$$\frac{\partial E(n)}{\partial w_o(n)} = \frac{1}{N} \sum_{j=1}^{N} e_j(n), \quad \text{(bias)}$$

(2) *Positions of centers* (hidden layer)

$$\frac{\partial E(n)}{\partial \boldsymbol{c}_i(n)} = \frac{2}{N} b_i(n) w_i(n) \sum_{j=1}^{N} e_j(n) o1_{ij}(n)(\boldsymbol{x}_j - \boldsymbol{c}_i)$$

(3) *Spreads of centers* (hidden layer)

$$\frac{\partial E(n)}{\partial b_i(n)} = -\frac{1}{N} w_i(n) \sum_{j=1}^{N} e_j(n) o1_{ij}(n) \|\boldsymbol{x}_j - \boldsymbol{c}_i\|^2$$

Notice that good performance is achieved only with 4 hidden nodes despite the fact that the original problem was constructed using 5 nodes. This is due to the capacity of the SS-EBM to avoid the local minima of the error surface. For this reason, together with the fact the initial training set is linearly separable, we argue that the SS-EBM may be used as a constructive mechanism in order to find the optimal number of hidden nodes in a neural network architecture.

From the simulations results, we observe that when SS-EBM is used to train both MLPs and RBFs by means of a gradient algorithm, the convergence rate increases due to the decreased probability of becoming stuck in a local minima.

Finally, we offer the following brief comments on the computational costs of using BMs. We have observed that both in the training of the MLP networks and in the training of the RBF networks the computational cost is reduced when BM is employed. This is a feature shared by several SS algorithms based on incremental learning (IL) procedures. While a formal analysis and complete experimental verification of the computational benefits of using incremental training is beyond the scope of this paper, we note that [30] provides an intuitive explanation of these computational profits.

4. Conclusions

In this paper we have shown that BM are a viable approach for estimating class separability, and also have significant utility when used to train NNs by means of gradient algorithms. We described the methodology behind BMs in order to establish a basis for the reported experimental results.

When BM are used as a class separability estimation technique, the experimental results show that boundary method's measure, OS, is related to Bayes' error, ε. In all cases, regardless of distribution, number of classes, number of modes, or dimensionality, the correspondence of OS and ε was maintained. We conclude that BMs are a useful and computationally attractive method of estimating the separability of a given data set. Our ongoing research is focused on developing a more rigorous set of analytical results which will establish the precise relationship between OS and ε.

We have also demonstrated the utility of the BM as a SS procedure to train NNs trained using gradient algorithms. The selected NNs we used include the classical MLP trained with the BP algorithm, and RBF trained with a gradient algorithm. All simulations show that when the BM method is used as a SS procedure, the decision boundaries are clearly better that those achieved using the gradient algorithm alone. Classical methods used to train NNs via gradient algorithms use all the available samples of the training set in an attempt to find a good local minimum of an complex error surface. Normally, the probability of being trapped in a poor local minimum is high, and depends to some degree on the initial value of the network's weights. However, when the BM procedure is used, the complexity surface of the problem is slightly modified at each step and a good local minimum corresponding to the current error surface is easily located. In the first step (tangential boundaries, ZVO) the error surface is simpler because the training set is small (it is only composed by the samples inside the tangential boundaries) and the problem becomes linearly separable. In particular, for the case of a MLP trained using batch mode BP, the convergence can be guaranteed. This fact makes the problem of determining the initial values of the network's weights inconsequential. In subsequent steps additional pattern samples are included in the training set, which modifies the error surface. In this fashion the network can more readily converge to a good local minimum. Similar results can be applied to RBF neural networks.

References

[1] A. Bhattacharyya, On a measure of divergence between two statistical populations defined by their probability distributions, Bull. Calcutt Math. Soc. 35 (1943) 99–110.
[2] M. Bianchini, P. Frasconi, M. Gori, Learning without local minima in radial basis function networks, IEEE Trans. Neural Networks 6 (1995) 749–756.
[3] M. Bianchini, M. Gori, Optimal learning in artificial neural networks: a review of theoretical results, Neurocomputing 6 (1996) 313–346.
[4] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, New York, 1995.
[5] C. Cachin, Pedagogical pattern selection strategies, Neural Networkds 7 (1) (1994) 175–181.

[6] C.H. Chen, On information and distance measures, error bounds and feature selection, Inform. Sci. 10 (1976) 159–173.
[7] H. Chernoff, A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations, Ann. Math. Statist. 23 (1952) 493–507.
[8] M.P. Clark, On the resolvability of normally distributed vector parameter estimates, IEEE Trans. Signal Process. 43 (12) (1995) 2975–2981.
[9] L. Devroye, A Course in Density Estimation, Birkhauser, Boston, MA, 1987.
[10] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley Interscience, New York, 1973.
[11] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd Edition, Academic Press, New York, 1990.
[12] S.I. Gallant, Neural Networks Learning and Expert Systems, MIT Press, Cambridge, MA, 1993.
[13] M. Gori, A. Tesi, On the problem of local minima in backpropagation, IEEE Trans. Pattern Anal. Mach. Intelligence 14 (1) (1992) 76–86.
[14] M. Gori, Ah Chung Tsoi, Comments on local minima free conditions in multilayer perceptrons, IEEE Trans. Neural Networks 9 (5) (1998) 1051–1053.
[15] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan College Publishing, Ontario, 1994.
[16] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, Reading, MA, 1991.
[17] C. Lee, D.A. Landgrebe, Feature extractoin based on decision boundaries, IEEE Trans. Pattern Anal. Mach. Intelligence 15 (4) (1993) 388–400.
[18] D.J.C. MacKay, Bayesian methods for adaptive models, Ph. D. Thesis, Caltech, Pasadena, CA, 1992.
[19] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Statist. 33 (1962) 1065–1076.
[20] W.E. Pierson, Using Boundary Methods for Estimating Class Separability, Ph. D. Thesis, The Ohio State University, Ohio, 1998.
[21] W.E. Pierson, B. Ulug, S.C. Ahalt, J.L. Sancho, A.R. Figueiras-Vidal, Theoretical and complexity issues for feature set evaluation using boundary methods, Proceedings of the 1998 SPIE Conference on Automatic Target Recognition VII, ATR Theory & Performance Estimation, Vol. 3070, 1998, pp. 173–184.
[22] M. Plutowski, H. White, Selecting concise training sets from clean data, IEEE Trans. Neural Networks 4 (1993) 305–318.
[23] D.E. Rumerlhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: D.E. Rumelhart, J.L. McClelland, the PDP Research Group (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Foundations, MIT Press, Cambridge, MA, 1986, pp. 318–362.
[24] J.L. Sancho, A.R. Figueiras-Vidal, B. Ulug, W. Pierson, S.C. Ahalt, Boundary methods for distribution analysis, Proceedings of the Fourth Bayona Workshop on Intelligent Methods for Signal Processing and Communications, Bayona (Vigo), Spain, June 1996, pp. 6–10.
[25] C.W. Therrien, Decision, Estimation, and Classification: An Introduction to Pattern Recognition and Related Topics, Wiley Interscience, New York, 1989.
[26] L. Vandenverghe, S. Boyd, Semidefinite programming, SIAM Rev. 38 (1996) 49–95.
[27] K. Yao, A representation theorem and its application to spherically-invariant random processes, IEEE Trans. Inform. Theory 19 (1972) 600–608.
[28] T.Y. Young, K.-S. Fu, Handbook of Pattern Recognition and Image Processing, Academic Press, Orlando, FL, 1986.
[29] B. Zhang, Focused incremental learning for improved generalization with reduced training sets, in: T. Kohonen, et al., (Eds.), Artificial Neural Networks: Proceedings of the ICANN, Vol. 1, Elsevier, Amsterdam, 1991, pp. 227–232.
[30] B. Zhang, Accelerated learning by active example selection, Int. J. Neural Systems 5 (1) (1994) 67–75.

José-Luis Sancho received his Physics degree from the Universidad de La Laguna, Tenerife (Spain) in 1992, his M.S in Electrical Engineering from the Universidad Politécnica de Madrid, Madrid (Spain) in 1994, and his Ph.D. in Electrical Engineering from the Universidad Carlos III de Madrid, Madrid (Spain) in 1999. Currently, he is an Assistant Professor at the Universidad Carlos III de Madrid, Madrid (Spain). His research interests include Digital Signal Processing, Statistical Pattern Recognition, Neural Networks, and Learning Theory.

Bill Pierson received his B.S. in electrical engineering from the West Virginia Institute of Technology in 1991 and his M.S. and Ph. D. in electrical engineering from The Ohio State University in 1993 and 1998 respectively. Currently, he is employed by the United States Air Force Research Laboratory (AFRL) at Wright Patterson Air Force Base located in Dayton Ohio. His work includes the evaluation of Automatic Target Recognition (ATR) systems and his areas of interests include pattern recognition theory, signal processing, computer vision, and information theory.

Batuhan Ulug received his BSEE from Bogazici University, Istanbul (Turkey) in 1990 and his MSEE from The Ohio State University in 1992. In the academic year 1990–1991 he was a University Fellow and from 1991 to the present he has been a Graduate Associate with the Department of Electrical Engineering at The Ohio State University. He is currently pursuing a Ph. D. His research interests are in pattern recognition, signal processing and neural networks. He is a member of the IEEE Information Theory Society.

Aníbal R. Figueiras-Vidal obtained his Telecomm Engineer degree from Universidad Politécnica de Madrid (Spain) in 1973, and his Doctor degree in 1976 from Universidad Politécnica de Barcelona (Spain). He is a Professor in Signal Theory and Communications at Universidad Carlos III de Madrid. His research interests are Digital Signal Processing, Digital Communications, Neural Networks, and Learning Theory.

Stanley C. Ahalt received his BSEE and MSEE degrees from the Virginia Polytechnic Institute and State University in 1978 and 1980 respectively. He obtained his Ph. D. in Electrical Engineering from Clemson University in 1986. Since 1987, he has been with the department of Electrical Engineering where he is currently a Professor in the Department of Electrical Engineering. During 1980 and 1981, Dr. Ahalt worked at Bell Telephone Laboratories where he developed industrial data products. Dr. Ahalt's research interests include neural networks, data compression algorithms, real-time video compression, and vector quantization, with applications to automatic target recognition, image compression, speech analysis, and robotics. Dr. Ahalt is a former Associate Editor of the IEEE Transactions on Neural Networks and he is a member of the International Neural Network Society and the Institute of Electrical and Electronics Engineers.