Brief Papers

Sample Selection Via Clustering to Construct Support Vector-Like Classifiers

Abdelouahid Lyhyaoui, Manel Martínez, Inma Mora, Maryan Vázquez,
José-Luis Sancho, Aníbal R. Figueiras-Vidal, *Senior Member, IEEE*

***Abstract*— This paper explores the possibility of constructing RBF classifiers which, somewhat like support vector machines, use a reduced number of samples as centroids, by means of selecting samples in a direct way. Because sample selection is viewed as a hard computational problem, this selection is done after a previous vector quantization: this way obtaining also other similar machines using centroids selected from those that are learned in a supervised manner. Several forms of designing these machines are considered, in particular with respect to sample selection; as well as some different criteria to train them. Simulation results for well-known classification problems show very good performance of the corresponding designs, improving that of support vector machines and reducing substantially their number of units. This shows that our interest in selecting samples (or centroids) in an efficient manner is justified. Many new research avenues appear from these experiments and discussions, as suggested in our conclusions.**

***Index Terms*— Classification, generalization, radial basis functions, clustering, sample selection.**

I. Introduction

THE theoretical basis of support vector machines (SVM's) is the application of structural risk minimization (SRM) using the Vapnik–Chervonenkis (VC) dimension [1]. Although first versions of SVM appeared in [2]–[4], [5] presents a more complete view of them. Since we will address here their application to classification, we will follow [4], [6]; many other aspects can be found in the extensive bibliography [7].

When considering the application of S samples to a binary single-layer perceptron (SLP) (a linear classifier), the SRM formulation requires the solution of

$$\min_{\boldsymbol{w}} C(\boldsymbol{w},\xi) = \min_{\boldsymbol{w}} \left(\tfrac{1}{2} \, ||\boldsymbol{w}||_2^2 + \gamma \sum_{s=1}^{S} \xi_s \right) \qquad (1)$$

where $\boldsymbol{w}$ are the SLP weights, γ a weighting parameter, and $\xi_s \geq 1 - d_s o_s$ slack parameters related to the desired output d_s and the SLP output o_s for each sample s. Note that minimizing the second term of (1) is equivalent to the minimization of the L_1-norm error if $\{|o_s|\}$ are required to be less than one: this is forced by the method applied to carry out such a minimization, quadratic programming (QP). Thus, this SVM approach is related to the ideas of Telfer and Szu [8], [9], who discovered that L_1-norm objective functions are adequate to get reduced misclassification rates if the weights are kept small. This requirement has also been related with generalization capabilities via the VC dimension in [10].

The use of a QP to solve (1) leads to selecting only a part of the samples to compute the SLP weights. These samples are called support vectors (SV's), and, qualitatively, they are the "key" samples to define the classification border. The above facts led Vapnik and coworkers to say that the other samples are "irrelevant" [6]. It must be remarked that the "irrelevance" of the discarded samples is more a result (due to the way of working of the QP) than a principle.

The above formulation was also applied in an extended way: for example, a "global" (Gaussian) radial basis function (RBF) classifier is proposed as a first structure, and, then, the above formulation is applied to its output layer (in fact,an SLP). The "global" network has an output

$$o_s = \sum_{s'=1}^{S} w_{s'} \exp\left(\frac{-||\boldsymbol{x}_s - \boldsymbol{x}_{s'}||_2^2}{2\sigma_{s'}^2} \right) + b \qquad (2)$$

where $\boldsymbol{x}_{s'}$ is the value of the s'th sample, and $\{w_{s'}\}, b$, the output layer weights.

When applying (1) to this construction, we find again that there are "irrelevant" samples; but an interpretation that opens more avenues to research is to see this as selecting a reduced architecture. In fact, this point of view connects SVM with an obvious precedent, weight pruning; and, in particular, with the "weight decay" procedure proposed by Hinton for multilayer perceptrons (MLP's) [11].

SV lie in the proximity of the classification border. This connects the SVM with sample selection (SS) methods in an implicit way. But, before going ahead in exploring the possibility of using SS to design SVM-like classifiers, let us dicuss some practical aspects of (standard) SRM-based SVM classifiers.

II. Some Limitations of (SRM-Based) SVM Classifiers

From an analytical/structural point of view:

1) There are some parameters to select: weighting value γ, and kernel "deviation" σ.
   Many experiments demonstrate that selecting γ is not (extremely) critical. Furthermore, since we may try different values of γ and select one of them according to the empirical results, this is not a limiting factor.

Manuscript received January 14, 1999; revised May 21, 1999. This work was supported in part by CICYT under Grant TIC96-0500-C10-03.

The authors are with the Department of Communication Technologies, Universidad Carlos III de Madrid, 28911 Leganés-Madrid, Spain.

Publisher Item Identifier S 1045-9227(99)07271-9.

In principle, the same could be said for σ; additionally, this parameter has been found to be not critical for the performance of RBF-based classifiers if the number of neurons is high [12] (discussing [13]). Furthermore, there are also rules to select σ before applying the SVM formulation [5]. However the resulting design appears with same σ for all the SV: it cannot be said that this is better than allowing different σ for different SV.

2) The number of SV is relatively high; and, even worse, in some cases (showing a relatively high degree of class overlapping) some "useless" pairs (one sample located at the wrong side of the border plus one sample of the opposite class very near to it) are selected. If other design procedures could show equivalent results with less elements, they would certainly have an advantage over SVM.

   (Note that we are not commenting on the high computational effort for designing standard SVM: in fact, it is high for the "block" SVM formulation we are considering here, but there are other formulations that require less [14]–[16].)

   And, with respect to operational aspects:

3) There is not an immediate extension of standard SVM formulation to multiclass problems. The multiple "one class against its anticlass" approach used in [6], for example, gives poor results in practice
4) There is not an efficient adaptive version of standard SVM designs. It is evident that to repeat the QP step deleting old and introducing new samples (or using any other approach to disregard the past and consider the present) is not efficient, and it is not possible to look for directly "adaptive" versions of the QP.

These limitations are not enough to conclude that standard SVM are poor classifiers: they have proved the contrary. The idea goes in just the opposite sense: standard SVM have such a high performance that alternative ways to construct similar machines, but avoiding some of the above limitations, merit particular attention. One of such alternatives is based in considering that the essential characteristic of SVM is its implicit SS.

III. Sample Selection-Based "SVM" Through Clustering

A simple approach to design a SVM based on SS will be to select a preliminary classifier, then samples near the border, and to build an RBF classifier using them. However, this is not the objective, but to take those samples that are the most important to define the border.

Many methods to select samples have been proposed: [17] giving an extensive list of possibilities. SS has also been applied to construct reduced RBF classifiers (or SS-based "SVM," according to our denomination) [18] by adding new selected samples according to their proximity to the class borders at each step. But, in our opinion, one of the first and (apparently) straightforward proposals [19] has the best conceptual framework to carry out SS. Munro says that it is better to apply more frequently the samples that are more difficult to learn. This means that these are the "critical" samples in order to solve the problem. However, the problem of selecting samples remains open, even in the light of [19]. Which are the "critical" samples will depend, in general, on the classifier being used, and an arbitrary preliminary classifier could introduce an unacceptable dependence on this preliminary selection.

One alternative is to reduce the size of the selection problem by means of clustering. This is not a new idea: some of Kohonen's LVQ [20] use a "window" to decide when to train a centroid, which is equivalent to saying that the critical samples are those located in the window. Following this idea, a procedure for SS is as follows.

1) Cluster the samples of each class.
2) Decide which clusters are critical (reducing the size of the selection problem).
3) Select samples in each cluster according to their "proximity" to the border (estimated by applying a cluster based classifier).

We remark that:

1) The suggested approach has the apparent drawback of calling for some additional parameters to be established, e.g., initial number of centroids (for each class), clustering parameters, number of selected samples per cluster, etc. All of these are known problems with a variety of solutions. On the other hand, note that these options are new "degrees of freedom" that can be useful to obtain better performance. Additionally, the parameters $\{\sigma_s\}$ can be obtained via unsupervised or supervised modes.
2) It is obvious that the proposed method allows an easy elimination of the "useless" pairs.
3) The procedure is directly applicable to multiclass problems.
4) Since clustering, determination of critical clusters, and subsequent SS can be easily made adaptive, we immediately obtain adaptive SVM-like structures.

   Additionally:

5) Any cost objective can be introduced to train the output layer weights.
6) There is the possibility of creating many "combined" classifiers: for example, we can use the SS, or SVM, ideas just for critical clusters, and the other (supervised) clusters to classify directly the samples which are included by them.

IV. A Particular SVM Based on Sample Selection by Means of Clustering

A. Clustering Step

In our experience, the clustering algorithm is not a critical element. Thus, we apply here frequency sensitive competitive learning (FSCL) [21], [22], an efficient scheme that avoids initialization problems. In particular, we apply FSCL assigning a number of centroids to each class (10, 15, 20, in the examples that will follow), and using a reinforcing/antireinforcing

training

$$\boldsymbol{c}_j(k+1) = \begin{cases} \boldsymbol{c}_j(k) + \beta(k)[\boldsymbol{x}_k - \boldsymbol{c}_j(k)], & x_k \in C_j \\ \boldsymbol{c}_j(k) - \beta(k)[\boldsymbol{x}_k - \boldsymbol{c}_j(k)], & x_k \neq C_j \end{cases} \quad (3)$$

where $\beta(k)$ is a contractive learning parameter that is reduced linearly from 0.3 to 0.15 from iteration 1–5000, from 0.15 to 0.05 for k going from 5001 to 8000, and to reach zero from step 8001–10 000 (this mode being empirically selected), C_j is the class of the centroid $\boldsymbol{c}_j(k)$, and the winner sample $\boldsymbol{x}_k$ is decided with the usual FSCL modified distance criterion

$$j = \arg\min_m \{\text{freq}_m(k)\|\boldsymbol{x}_k - \boldsymbol{c}_m(k)\|_2^2\}. \quad (4)$$

B. Supervised Learning for Clustering

Kohonen's LVQ3 [20] is applied in the following form:

1) if the two closest centroids to sample $\boldsymbol{x}_k$ are $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$, $\boldsymbol{c}_i$ corresponding to $\boldsymbol{x}_k$'s class and $\boldsymbol{c}_j$ corresponding to a different class

$$\boldsymbol{c}_i(k+1) = \boldsymbol{c}_i(k) + \gamma[\boldsymbol{x}_k - \boldsymbol{c}_i(k)] \quad (5a)$$

$$\boldsymbol{c}_j(k+1) = \boldsymbol{c}_j(k) - \gamma[\boldsymbol{x}_k - \boldsymbol{c}_j(k)] \quad (5b)$$

where $\boldsymbol{x}_k$ is in a 20%-width window as defined in [20];

2) if the two closest centroids are of the same class than $\boldsymbol{x}_k$

$$\boldsymbol{c}_{i,j}(k+1) = \boldsymbol{c}_{i,j}(k) + \delta[\boldsymbol{x}_k - \boldsymbol{c}_{i,j}(k)]; \quad (5c)$$

3) no changes in other cases.

The algorithm is stopped after 10 000 steps. Values of γ and δ are empirically fixed at 0.02 and 0.04, respectively. The stopping instant is also found empirically.

C. Selection of Critical Centroids

All the centroids are visited, and, for each one, the nearest centroid of the other class is determined. When two centroids of different classes are the nearest in both senses, both are included in the first group of critical centroids (as done in [23]). A refinement process is added to the above first selection in order to include other centroids being important to define the border. The selected centroids are used to classify the remaining centroids according to a 1-NN (nearest neighbor) rule; among the wrongly classified centroids, that being closer to any (previously) selected centroid is transferred to the set of selected centroids. The process is iterated until there are no errors.

Our experimental work shows that this is the most important step of the construction procedure. There are other reasonable alternatives for carrying it out, but we have followed the above approach because it is straightforward and efficient. The apparently relative high computational load can be easily reduced using short cuts similar to those shown in [24] for NN classifiers.

D. Computing Dispersion Parameters for the Centroids

Since we are selecting centroids and there is the possibility of taking advantage of additional parameters (dispersion values and output weights) in a Gaussian RBF scheme, using selected centroids to create an RBF classifier is an attractive methodology (although it cannot be strictly considered an SVM). On the other side, some of the computations required here will be needed in further steps to construct SS-based "SVM."

The first question is how to establish dispersion parameters $\{\sigma_j\}$ for the selected centroids. In principle, they can be computed by applying an error-gradient training, but it is well known that doing so produces serious difficulties due to local minima. It seems reasonable to admit that the values $\{\sigma_j\}$ must be proportional to the distance to the classification border of the corresponding centroid, to allow the corresponding neuron to have a preferential action on the samples in its cluster and located on the same side of the border. However, it is not clear what "scale" factor must be convenient. We choose to search for δ and d_j separately, to find $\sigma_j = \delta d_j$, d_j being related to the distance of $\boldsymbol{c}_j$ to the border, and δ being a (general) scale parameter.

In principle, d_j is selected as half the distance between $\boldsymbol{c}_j$ and the nearest centroid $\boldsymbol{c}_i$ of the opposite class. But there are cases in which $\boldsymbol{c}_j$ can be the nearest centroid of the opposite class also for $\boldsymbol{c}_l \neq \boldsymbol{c}_i$. In these cases, $d_l = d_j$ (the minimum value) is selected. This modification tends to keep the (preliminary) local border given by the LVQ design in the same position, as would be reasonable.

Now, we discuss how to obtain a reasonable initial value for δ. First, consider the case of a two kernel network for an one-dimensional problem. The output will be

$$o = w_1 \exp\left[-\frac{(x-c_1)^2}{2(\delta d)^2}\right] + w_2 \exp\left[-\frac{(x-c_2)^2}{2(\delta d)^2}\right] + b \quad (6)$$

with $c_1 = -c_2 = d$. Assuming that the final trained network will be near the LVQ solution, $w_1 \simeq -w_2 \simeq 1$ and $b = 0$ are acceptable weights ($x = 0$ is the solution). The sensitivity of output o (of the decision) with respect to the border is

$$s = \left.\frac{\partial o}{\partial x}\right|_{x=0} = 2\frac{\exp(-1/2\delta^2)}{\delta^2 d} \quad (7)$$

and it seems reasonable to minimize its variation with respect to δ

$$\frac{\partial s}{\partial \delta} = 2\left(\frac{1}{\delta^2} - 2\right)\frac{\exp(-1/2\delta^2)}{\delta^3 d} = 0 \quad (8)$$

from which $\delta_0 = \sqrt{2}/2$.

Fig. 1 shows o versus x for δ_0, $2\delta_0$, and $\delta_0/2$. s rapidly decreases for $\delta < \delta_0$, and also decreases for $\delta > \delta_0$, but in a slower fashion. To have a flat output around x is not convenient, because small changes in the weights will change the border considerably.

To extend the above to N dimensions is simple. The only change is, forcing $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ lying on axis x_i

$$\|\boldsymbol{x} - \boldsymbol{c}_i\|^2 = \|\boldsymbol{x}|_{x_i=0}\|^2 + d^2 = k_{\boldsymbol{x}}^2 + d^2 \quad (9)$$

from which, in the same way as in the above case

$$\delta_{o\boldsymbol{x}} = \frac{\sqrt{2(k_{\boldsymbol{x}}^2 + d^2)}}{2d}. \quad (10)$$

Selecting for $k_{\boldsymbol{x}}^2$ a value $k_{\boldsymbol{x}_0}^2$ from $\sqrt{k_{\boldsymbol{x}_0}^2 + d^2} = 2d$ is enough to guarantee that (assuming approximate Gaussianity)

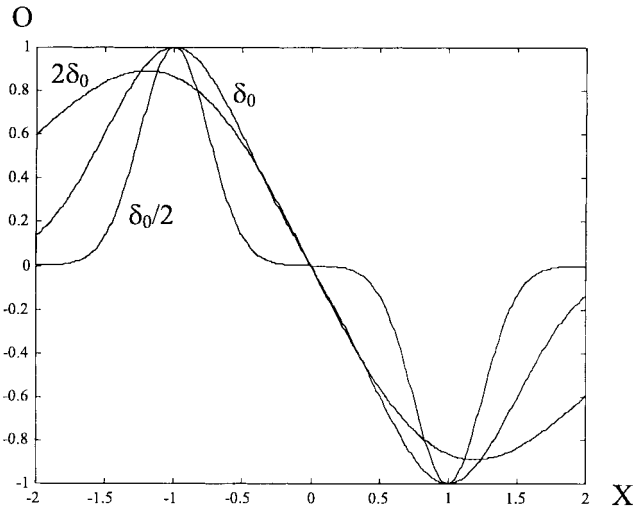

Fig. 1. Output of the two-element network versus x with respect to parameter δ: δ_0 is optimal in the sense of resulting in a maximum slope at $x = 0$.

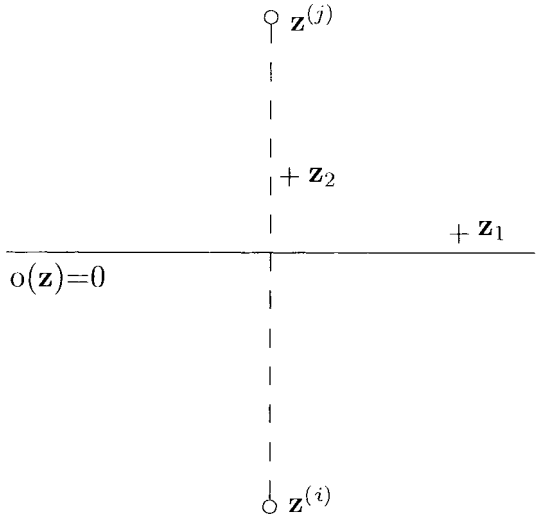

Fig. 2. Illustrating the problem of selecting z_1, nearer to the border, and not z_2, more "typical." $z^{(j)}$, $z^{(i)}$, are the images of the relevant centroids in the intermediate space.

most the samples in the cluster around $\boldsymbol{c}_i$ give a value for $\delta_0 \boldsymbol{x}$ bounded by $\delta_0 \boldsymbol{x}_0 = \sqrt{2}$ that corresponds to such a selection (note that the difference with respect to the one-dimensional case is not great).

E. Training $\boldsymbol{w}, b$, and $\{\sigma_j\}$ for the SC-Based RBF Classifier

$\{w_j\}$ are initialized at values ± 0.1 (sign according to the corresponding class), in order to keep their values small, if possible; the initial value of b is zero. Conventional gradient type algorithms are applied using an error-rate based cross-validation to select trained values. The same procedure is done for δ, in the cases in which δ is trained.

There is the possibility to select any objective functional for the final training. We have examined standard L_2, L_2 using a hyperbolic tangent output, L_1 with the same nonlinear unit, and the entropic objective proposed in [25] and [26] (that also requires a hyperbolic tangent output). These objectives are indicated by LE2, SE2, SE1, and ENT, respectively, in the following.

Maximum saturation level has been fixed at 0.99 for the nonlinear cases.

F. Selecting Samples

In principle, it seems that, since we have local problems, to select the samples nearest to the border can be enough, since criticality has been considered when selecting centroids. But if one uses $o(\boldsymbol{x}) \simeq 0$ (positive or negative according to the class) for this selection, o being the output corresponding to an associated (and previously designed) SC-based RBF classifier, although o is an indicator of the proximity to the border, some problems appear. These problems are related to the insufficiency of the output values to carry out an adequate selection, as empirically recognized in previous references of sample selection and explicitly discussed in [27].

Proximity to the border is important, but it does not consider if the sample is "typical" or not, in the sense that Fig. 2 shows. In the intermediate space $\boldsymbol{z}$ (outputs of RBF elements), $\boldsymbol{z}_1$ is nearer to the border than $\boldsymbol{z}_2$: but, since a clustering process has been applied, it is more likely that other samples (including new ones) be nearer to $\boldsymbol{z}_2$ than to $\boldsymbol{z}_1$, because $\boldsymbol{z}_1$ is further from its "centroid" than $\boldsymbol{z}_2$. It is also easy to visualize the effect of selecting $\boldsymbol{z}_1$ (as any other "atypical" sample): taking it as a centroid will have the effect of a tendency to move the border away it only because its atypical position.

It is evident that to select many samples for each centroid can reduce this difficulty, but this would unnecessarily increase the size of the resulting classifier. We follow another less computationally expensive solution: selecting samples considering also their "typicalness," assuming that a sample is more "typical" if it is "nearer" to the projection of the image of $\boldsymbol{c}_j$ on the border of the intermediate space.

Projecting $\boldsymbol{z}^{(j)}$ into the border $\boldsymbol{w}^T\boldsymbol{z} + b = 0$ ($\boldsymbol{w}, b$, corresponding to the SC-based RBF classifier) is easy. Consider the straight line $\boldsymbol{z} = \lambda \boldsymbol{w} + \boldsymbol{z}^{(j)}$, which is orthogonal to the border and contains point $\boldsymbol{z}^{(j)}$. Introducing this value of $\boldsymbol{z}$ into the equation of the separating hyperplane, the value of λ for the projected point is

$$\lambda = -\frac{\boldsymbol{w}^T\boldsymbol{z}^{(j)} + b}{||\boldsymbol{w}||_2^2} \tag{11}$$

which gives the projected point as

$$\boldsymbol{z}_0^{(j)} = \boldsymbol{z}^{(j)} - \frac{\boldsymbol{w}^T\boldsymbol{z}^{(j)} + b}{||\boldsymbol{w}||_2^2}\,\boldsymbol{w}. \tag{12}$$

Now, we select samples according to the values of the mixed indicator (proximity to the border plus typicalness)

$$I[\boldsymbol{z}(\boldsymbol{x})] = |\boldsymbol{w}^T\boldsymbol{z} + b| + |\boldsymbol{z} - \boldsymbol{z}_0^{(j)}| \tag{13}$$

(always considering samples located at the correct side of the border). The use of L_1 measures is simple and does not reduce performance.

How to select more than one sample? It is evident that to select a series of samples very near one to the other is not a good policy; then, to use just (13) is not sufficient. Thus, some

"dispersion" is forced by requiring, via an iterative mode of selection, not only that the last selected sample gives a value of I lower than the previously explored samples, but also that its distance to its centroid be lower than its distance to any of the previously selected samples. This method will be called direct selection (DS).

What about situations in which selected samples are not "balanced" with respect to the line connecting the images of the centroids in the intermediate space? Following the discussion of "typicalness," this will modify the decision border in an undesired manner. To avoid this, we propose an alternative that we will call selecting pairs (SP's). It consists of appending to DS a second phase: for each selected sample, the nearest (correctly classified) sample of the other class is also selected, if it has not been selected by DS.

What about situations in which a centroid is surrounded by a group of the other class? It is clear that DS and SP are "suboptimal" for these situations. One method we have developed for these cases is alternative selection (AS): a DS phase is applied to one class, and, after it, the SP procedure is used; finally, DS is applied to the other class to complete the specified number of samples to be selected for each centroid.

A final method, a little bit different, which will be called pair selection (PS), is as follows.

1) For each centroid of one of the classes, the nearest centroid of the other class is found.
2) Samples are selected according to their proximity to the middle point between these centroids (among correctly classified samples).
3) For each centroid of the second class not selected in the above process, the process is repeated in the same form.

There are many other (reasonable) forms to select samples: however, we think that the above four are basic representatives of most of them.

G. Computing Dispersion Parameters for the Selected Samples

The basic decomposition $\sigma'_k = \delta d'_j$ is also proposed for this case: however, there is not a clear argument to establish δ'_0.

A qualitative argument is the following: let us assume that $2d'$ is the distance between two selected samples, both being near to the border. Using the same reasoning as for centroids, the only difference will be the value we need for $k_{\boldsymbol{x}}$. But $k_{\boldsymbol{x}}$ is qualitatively equivalent to its counterpart for the centroids: in that case, it was bounded by $k_{\boldsymbol{x}_0} = \sqrt{3}d$. So, disregarding d' in the numerator of $\delta'_{0\boldsymbol{x}}$, we have $\delta'_0 = \sqrt{3/2}(d/d')$. In the experiments, we have used an empirical value $d/d' = 3(\delta'_0 = 3\sqrt{3/2})$.

H. Training $\boldsymbol{w}, b$, and $\{\sigma_j\}$ for the SS-Based RBF Classifiers

As for the SC-based RBF classifiers, but the δ' searches are different according to SS.

V. Some Experiments and Their Discussions

A. Ripley's Dataset

We have 250 training and 1000 testing samples per class in a binary two-dimensional problem corresponding to double-Gaussian class distributions with a high degree of overlapping.

TABLE I
Results of Applying SC-Based Classifiers with Theoretical $\delta_0(\simeq 1.42)$ to Ripley's Data (Percentage of Correct Classifications) for Different Cost Functions: Both the Average Performance for the Ten Initializations ($\overline{cc}$) (Plus Standard Deviation) and the Best Performance (cc_m) Are Shown. Subindexes 10, 15, and 20 Refer to the Three Numbers of Clusters

	$\overline{cc}_{10}$	cc_{10m}	$\overline{cc}_{15}$	cc_{15m}	$\overline{cc}_{20}$	cc_{20m}
LE2	89.7 (2.1)	91.5	89.6 (2.6)	91.0	88.8 (0.7)	90.8
SE2	89.6 (2.1)	91.6	89.7 (2.9)	91.4	89.7 (1.5)	90.8
SE1	89.5 (2.1)	91.8	89.7 (2.8)	91.4	89.6 (2.4)	90.9
ENT	89.5 (2.2)	91.6	89.8 (2.6)	91.4	89.0 (3.1)	91.2

Its original source is [28]. This data have been also used in [29] for (modified) standard SVM (data available at ftp://markov.states.ox.ac.uk/pub/neural/papers/synth.data and ftp://markov.states.ox.ac.uk/pub/neural/papers/synth.test).

First of all, we apply standard SVM to this problem, but allowing several values for σ and γ. A margin $\epsilon = 10^{-3}$ is used in the QP algorithm. Best results (90.0% correct classifications) are obtained for $\sigma = 0.25$ and 0.5, $\gamma = 100$, and $\sigma = 0.5$, $\gamma = 1000$ (numbers of SV equal to 73, 73, and 72, respectively), with a gentle degradation when moving away from these values. Results are comparable to those of [29].

Three families of SC-based RBF classification experiments are carried out. We start with $10 + 10$, $15 + 15$, and $20 + 20$ centroids (initialized uniformly around the average of each class, with a variance 0.01), and repeating ten times the process for each family. The training set is reduced to 212 ($106 + 106$) samples, the remaining 38 ($19 + 19$) being used for cross-validation when searching for final parameter values ($\boldsymbol{w}, b$, and, in some cases, δ) in order to obtain good generalization capabilities.

Table I shows the results obtained using the theoretical value δ_0: in this case, $\boldsymbol{w}$ and b are found by applying instantaneous gradient algorithms for each cost objective with a step of 10^{-4}, and the validation set is used to keep an error count per epoch while training. We stop when the cost, averaged over ten epochs, does not decrease more than 10^{-3}, and, then, the parameter values corresponding to the most recent minimum in the validation error count are selected. The particular value of the adaption step does seem not to be critical.

Note that we get better (peak) performance than that given by the standard SVM (to select the best design is usual when dealing with classifiers). We also remark that the average (standard deviation) for the resulting number of centroids are 6.8 (1.4), 9.0 (1.4), and 9.5 (0.8) for the $10 + 10$, $15 + 15$ and $20 + 20$ cases, respectively: thus, we are obtaining very good results with very simple (both from structural and learning points of view) machines! In particular, compare numbers of centroids with the above mentioned more than 70 samples for the SVM.

The ratio $\sigma_{\max}/\sigma_{\min}$ corresponding to this design and simulations has an average of 1.26 with a standard deviation of 0.12. We will discuss later these results.

Another interesting observation is that applying different cost functions does not greatly change the results. This can be

TABLE II
RESULTS OF APPLYING SC-BASED RBF CLASSIFIERS INCLUDING A GRADIENT SEARCH FOR δ TO RIPLEY'S DATA FOR DIFFERENT COSTS: CASES AND MAGNITUDES AS ABOVE, PLUS AN INDICATION ON THE (AVERAGE) FINAL VALUE OF δ, $\overline{\delta}_f$

	$\overline{cc}_{10}$	cc_{10m}	$\overline{\delta}_{f10}$	$\overline{cc}_{15}$	cc_{15m}	$\overline{\delta}_{f15}$	$\overline{cc}_{20}$	cc_{20m}	$\overline{\delta}_{f20}$
LE2	91.3 (0.5)	92.1	2.1 (0.8)	91.5 (0.4)	92.2	2.8 (0.5)	91.4 (0.5)	92.4	3.0 (0.4)
SE2	91.4 (0.6)	92.1	2.1 (0.7)	91.6 (0.3)	92.1	2.6 (0.8)	91.4 (0.6)	92.3	3.0 (0.6)
SE1	91.1 (0.6)	91.8	2.3 (0.8)	91.6 (0.5)	92.2	2.6 (0.7)	91.5 (0.7)	92.3	2.9 (0.5)
ENT	91.3 (0.5)	91.9	2.3 (0.6)	91.7 (0.5)	92.4	2.7 (0.7)	91.0 (1.9)	92.3	3.0 (0.5)

apparently surprising, but it is definitely a clear side effect of selecting centroids.

Of course, it is easy to conclude that these results are not the best because (among other factors) values σ_j were fixed under a "soft" empirical rule (to select δ). To verify the importance of this factor, a second type of experiments is carried out including a gradient-type search for δ. In particular, 20 equidistant values in the interval $0.5 - 5$ (1/3 to three times δ_0, roughly speaking) are used to initialize the algorithms, and the best result (considering the averages for the ten realizations) among all the 20 are selected. Table II gives the results for the case of a search step of 10^{-4} for δ.

These results correspond to RBF machines having the same (average) number of centroids as before, and they are excellent from any point of view (theoretical cc for Ripley's data is 92%! Note that cc_m values over 92 are not against the laws of statistics, since there is a sampling effect). The need for searching δ does not diminish the validity of the approach, since δ_0 provides a good reference value to define the search interval.

We can also give some measurements that support our conjecture that allowing different RBF dispersion parameters is important to get good classification results: $\sigma_{\max}/\sigma_{\min}$ values have average and standard deviation equal to 1.3 (0.1), 1.7 (0.5), and 1.8 (0.5), for the 10 + 10, 15 + 15, and 20 + 20 families of experiments, respectively, clearly different from 1. We have verified that equalizing these values leads to worse performance.

A final overall comment: the suggested approach not only offers advantageous results with respect to standard SVM, but also a low sensitivity with respect to design parameters, such as the initial number of centroids.

With respect to results of SS-based RBF classifiers, we will present thems for only the L_1 norm (differences are not important) and the 20 + 20 family of experiments. Conditions are as follows.

1) Two samples are selected when using DS (only one is not enough for acceptable results).
2) For SP, an 1 + 1 selection is enough (i.e., to start with an one-sample DS).
3) The AS method is applied until we have completed at least one sample per centroid (one pass).
4) Finally, PS is applied twice (so, selecting at least one sample per centroid).

Using δ_0' not only provides relatively poor performance, but yields different results according to the method to select samples. This is not unexpectable since the way to establish δ_0' was even more "approximate" than that for defining δ_0, but it is also obvious that different sample selection schemes greatly change their relative distances. However, the proposed δ_0' is also a good value for initializing searches on δ'. We have carried out these searches in the same way as in SC-based RBF classifiers, but with initial values separated 0.2 in the intervals (2, 6) for DS, (4, 8) for SP, (4, 8) for AP, and (2, 6) for PS.

TABLE III
RESULTS OF APPLYING SS-BASED RBF CLASSIFIERS TO RIPLEY'S DATA FOR SIGMOIDAL OUTPUT AND L_1 ERROR, 20 + 20 INITIAL CENTROIDS, ACCORDING TO THE DIFFERENT SAMPLE SELECTION PROCEDURES. PARAMETERS AS IN THE ABOVE TABLES

	$\overline{cc}$	cc_m	$\overline{\delta'_f}$
DS	91.0 (0.4)	91.4	3.1 (0.2)
SP	90.8 (0.5)	91.6	5.4 (0.2)
AS	90.7 (0.9)	91.2	5.5 (0.2)
PS	91.2 (0.3)	91.7	5.3 (0.6)

Table III shows results corresponding to the different sample selection schemes. The average (standard deviation) of number of selected samples is 16.8 (3.9) for DS and SP, 13.1 (3.6) for AS, and 17.4 (3.9) for PS, which are clearly lower than the number of SV for classical SVM designs, while obtaining better performance. This supports our claim of reduced operational computations and better results.

Performances in Table III are slightly worse than those corresponding to the above discussed SC-based RBF classifiers. The concept of "criticality" seems to be more important that the implicit effect of sample selection included in standard SVM.

$\sigma_{\max}/\sigma_{\min}$ is once again clearly different from unity: its values have an average (typical deviation) of 4.8 (2.9) for DS, 4.9 (1.9) for SP, 6.0 (2.3) for AS, and 4.4 (1.6) for PS, showing again that to allow these variations is important. Note also that the higher values for AS are associated with a reduced number of selected samples. We have verified that equalizing $\sigma_{\max}/\sigma_{\min}$ reduces performance.

It can be said, according to Table III, that there are not important differences between the above methods to select samples. This does not mean that how to select samples is irrelevant, but that there are many usable possibilities that serve for this purpose.

B. Other Experiments

We have applied also the above algorithms to the "Heart Disease" and the "Credit Screening" data available at PROBEN1, http://www.ai.univie.ac.at/oefai/ml/ml-

resources.html. These examples have been considered in [30] (with a 77% of correct classifications) and in [31], respectively.

Standard SVM peak results are obtained for $\sigma = 0.25$, 0.5, and 1, $\gamma = 100$, for the first case (with 331, 270, and 279 centroids, respectively), with 80.9% correct classifications, and for $\sigma = 0.5$, 1, and 2, $\lambda = 100$, and $\sigma = 1$, $\lambda = 1000$, for the second (number of centroids: 202, 283, 182, and 189, respectively), with 87.2% correct decisions.

While applying our SC-based RBF classifiers to "Heart" data, the best average value is 80.2% (1.9) correct classifications for SE2 criterion and $10 + 10$ centroids, and the peak, 82.2% in the same case with respect to centroids and for any error criterion; needing an (average) value of elements equal to 7.8. With SS-based RBF classifiers for $20 + 20$ centroids, the best average is 80.3% (1.5) for PS, and the best peak 82.6% for SP.

In the case of "Credit" data, the best average SC-based RBF classifiers are $20 + 20$ LE2 with 88.3% (1.1) correct classifications; the highest peak performance corresponds to this and other cases, providing 90.1% good decisions. 14.1 is the average number of centroids. DS is the best selection for SS-based RBF classifiers, with an average of 88.4% (1.0) and a peak of 89.5% correct classifications.

These results support our general comments in the previous example.

It is worth to mention that, in these cases, the efficiency of using theoretical $\delta_{\max}/\delta_{\min}$ is not as clear as for the Ripley dataset; in fact, we have checked that convex combinations of selected samples and their centroids give better results.

VI. Conclusions

We have investigated using sample selection to construct RBF-type classifiers, with an intermediate VQ to reduce the computational burden of such a selection. This idea allows us to obtain very efficient SVM-like machines, as verified with the particular approaches and examples studied in this paper.

In particular, it is remarkable that this methodology allows us to avoid the selection of undesirable (wrongly classified) samples, and to create RBF with different dispersion parameters. These facts seem to be essential to obtain moderate size (RBF-type) classifiers with very high performance, even in the case of stopping the designs at the VQ phase (including a centroid selection and an adequate training of free parameters, of course). It is also remarkable that different methods of sample selection do not change the results greatly. The same phenomenon occurs with respect to the training criteria being used: centroid or sample selection reduce the importance of the used criteria, as expected.

Many extensions are possible following this line of work, and, in particular, it is worth mentioning that local and hybrid (combined with other classifier) machines can be easily designed, and that extensions to multiclass problems and adaptive schemes are immediate.

Acknowledgment

The authors wish to thank Dr. V. Vapnik for useful discussions held during his visit to Universidad Carlos III, and to Prof. B. D. Ripley for orienting our attention to the paper in which he first used his database.

References

[1] V. Vapnik, *Estimation of Dependences Based on Empirical Data.* New York: Springer Verlag, 1982 (translated from Russian).
[2] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Wkshp. Comput. Learning Theory*, Pittsburgh, PA, 1992, pp. 144–152.
[3] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
[4] B. Schölkopf, C. Burges, and V. Vapnik, "Extracting support data for a given task," in *Proc. 1st Int. Conf. Knowledge Discovery and Data Mining*, U. M. Fayyad and R. Uthurusamy Eds. Menlo Park, CA: AAAI Press, 1995, pp. 252–257.
[5] V. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer Verlag, 1995.
[6] B. Schölkopf, K.-K. Sung, C. J. C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussians kernels to radial basis function classifiers," *IEEE Trans. Signal Processing*, vol. 45, pp. 2758–2765, 1997.
[7] A. Smola, B. Schölkopf, *Publications on Support Vector Machines and Related Topics*, Inst. Comp. Arch. Software Technology, German Nat. Res. Center Inform. Technol. Available www.first.gmd.de
[8] B. A. Telfer and H. H. Szu, "Implementing the minimum-misclassification-error energy function for target recognition," *Proc. Int. Joint Conf. Neural Networks*, vol. I, Baltimore, MD, 1992, pp. 214–219.
[9] ——, "Energy functions for minimizing misclassification error with minimum-complexity networks," *Neural Networks*, vol. 7, pp. 809–818, 1994.
[10] P. L. Bartlett, "For valid generalization, the size of the weights is more important than the size of the network," in *Advances Neural Inform. Processing Syst. 9*, M. C. Mozer, M. I. Jordan, T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, pp. 134–140.
[11] G. E. Hinton, "Learning translation invariant recognition in a massively parallel network," *Proc. Conf. Parallel Architectures Languages Europe*, Eindhoven, The Netherlands, 1987.
[12] S. Haykin, *Neural Networks. A Comprehensive Foundation.* New York: Macmillan, 1994.
[13] D. Lowe, "Adaptive radial basis function nonlinearities, and the problem of generalization," in *Proc. 1st IEE Int. Conf. Artificial Neural Networks*, London, U.K., 1989, pp. 171–175.
[14] C. Burges, "Simplified support vector decision rules," in *Proc. 13th Intl. Conf. on Machine Learning*, Bari, Italy, 1996, pp. 71–77.
[15] C. Burges and B. Schölkopf, "Improving the accuracy and speed of support vector machines," in *Advances in Neural Inform. Proc. Syst.*, vol. 9, M. Mozer, M. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, pp. 375–381.
[16] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *Proc. Neural Networks for Signal Processing*, Amelia Island, FL, 1997, pp. 276–285.
[17] C. Cachin, "Pedagogical pattern selection strategies," *Neural Networks*, vol. 7, pp. 171–181, 1994.
[18] E. I. Chang and R. P. Lippmann, "A boundary hunting radial basis function classifier which allocates centers constructively," in *Advances Neural Inform. Processing Syst. 5*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds. San Mateo, CA: Morgan Kaufmann, 1993, pp. 139–146.
[19] P. W. Munro, "Repeat until bored: A pattern selection strategy," in *Advances in Neural Information Proc. Sys. 4*, J. E. Moody *et al.*, Eds. San Mateo, CA: Morgan Kaufmann, 1992, pp. 1001–1008.
[20] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1480, 1990.
[21] S. Ahalt, A. K. Krishnamurty, P. Chen, and D. Melton, "Competitive learning algorithm for vector quantization," *Neural Networks*, vol. 3, pp. 277–290, 1990.
[22] S. Ahalt and J. E. Fowler, "Vector quantization using artificial neural network models," in *Adaptive Methods and Emerging Techniques for Signal Processing and Comms.*, D. Docampo, A. R. Figueiras-Vidal, Eds. Vigo, Spain: Universidad de Vigo, 1993., pp. 42–61.
[23] J. Sklansky and L. Michelotti, "Locally trained piecewise linear classifiers," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 2, pp. 101–111, 1980.
[24] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," *IEEE Trans. Comput.*, vol. 24, pp. 750–753, 1975.

[25] J. J. Hopfield, "Learning algorithm and probability distributions in feedforward and feedback networks," *Proc. Nat. Academy Sci. USA*, vol. 84, pp. 8429–8433, 1987.
[26] G. E. Hinton, "Connectionist learning procedures," *Artificial Intell.*, vol. 40, pp. 185–234, 1989.
[27] J. Cid-Sueiro, J. I. Arribas, S. Urbán-Muñoz, and A. R. Figueiras-Vidal, "Cost functions to estimate *a posteriori* probabilities in multiclass problems," *IEEE Trans. Neural Networks*, vol. 10, pp. 645–656, 1999.
[28] B. D. Ripley, "Neural networks and related methods for classification (with discussion)," *J. Roy. Statist. Soc. Series B*, vol. 56, pp. 409–456, 1994.
[29] E. Osuna and F. Girosi, "Reducing the run time complexity of support vector machines," *Proc. ICPR'98*, Brisbane, Australia, 1998.
[30] R. Dentrano *et al.*, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *Amer. J. Cardiology*, vol. 64, pp. 304–310, 1989.
[31] J. Quinlan, "Simplifying decision trees," *Int. J. Man-Machine Studies*, vol. 27, pp. 221–234, 1987.