

A Dynamically Adjusted Mixed Emphasis Method for Building Boosting Ensembles

Vanessa Gómez-Verdejo, Jerónimo Arenas-García, *Member, IEEE*,
and Aníbal R. Figueiras-Vidal, *Senior Member, IEEE*

Authors are with the Dept. of Signal Theory and Communications, Universidad Carlos III de Madrid, 28911 Leganés, Spain (e-mail: {vanessa,jarenas,arfv}@tsc.uc3m.es). Their work was partly supported by MEC Pjt. TEC2005-00992 and CM Pjt. S-0505/TIC/0223. The work of V. Gómez-Verdejo was also supported by the Chamber of Madrid Community and European Social Fund by a scholarship.

Abstract

Progressively emphasizing samples that are difficult to classify correctly is the base for the recognized high performance of Real Adaboost (RA) ensembles. The corresponding emphasis function can be written as a product of a factor that measures the quadratic error and a factor related to the proximity to the classification border; this fact opens the door to explore the potential advantages provided by using adjustable combined forms of these factors. In this paper, we introduce a principled procedure to select the combination parameter each time a new learner is added to the ensemble, just by maximizing the associated edge parameter; calling the resulting method the Dynamically adapted Weighted emphasis RA (DW-RA). A number of application examples illustrates the performance improvements obtained by DW-RA.

I. INTRODUCTION

An avenue that promises important benefits is the study of methods to combine Artificial Neural Networks (ANNs). Combinations of ANNs allow an easier design of classifiers and estimators, better performance, and even a clearer understanding of the reasons of the obtained outputs. A lot of work has been done along this avenue, as it can be seen in [1], [2].

Opposite to committee design, in which different ANNs are applied to solve (basically) the same problem under conditions that introduce some diversity (different sub-populations, architectures, initializations, etc.), the “cooperative” approaches try to construct ensembles by means of using ANNs that pay attention to the activity of the other members of the ensemble. One of these approaches is the modular architecture proposed by Jacobs and Jordan [3], [4], in which the outputs of several “experts” are combined according to the state of a “gate” network. The other main cooperative design procedure is boosting [5], [6], [7]. In particular, “confidence-rated Adaboost” [8], to which we will refer in this paper as Real Adaboost (RA), offers excellent performance for classification tasks¹.

Boosting is (apparently) based on combining base learners in a sequence in which each added machine pays particular attention to the samples showing a large classification error, following an idea that is fre-

¹Note that term Real Adaboost is not used in this paper in the same sense as in [9].

quently used in training conventional ANNs, consisting on emphasizing erroneous samples (see, among others, [10]-[15]). The alternative point of view suggested in [16] and further explored in [17], namely that boosting progressively concentrates its attention in samples nearer and nearer to the classification border, opens new possibilities for designing boosting mechanisms, that are also supported by the knowledge of the effectiveness of considering the proximity to the border for sample selection (see [18]-[20], for example). Since Breiman remarks [21] that the particular mode of emphasizing erroneous samples is not essential in order to get boosting benefits, the idea of paying attention to both the error and the proximity of each sample to the border, seems to be reasonable and potentially useful. We must remark that this concept is very similar to that of Maximum Separation Margin (MSM) [22] that, coming from Vapnik's work [23], has become the base of successful Support Vector Machines (SVMs) and kernel based classifiers [24], and has been applied [25] to surpass the limitations of a well-known method for growing ANNs, the Cascade Correlation procedure [26]. Even more, boosting ensembles, and in particular the RA algorithm, have been related to SVMs [27], [28], emerging new versions of this algorithm to get MSM solutions [29], [30].

After proving that the standard Adaboost resampling function can be decomposed into a product of two factors, the first depending on the quadratic error and the second taking into account the proximity of each sample to the classification border, we have generalized the structure of this product, introducing an adjustable mixing parameter to control the trade-off between both emphasis terms. In many cases, selecting the mixing parameter by means of cross-validation provides improvements with respect to standard RA schemes [31], but it is far from getting all the potential of the idea. Here, considering as an indication of the learner quality a generalized version of the learner edge (a weighted correlation between the learners outputs and the labels), we propose to adjust the mixing parameter by iteratively selecting the value that provides the learner with the largest generalized edge. This new dynamic procedure facilitates an adequate selection of the mixing parameter during the growth of the ensemble, providing better results

than the cross-validation approach.

The rest of the paper has the following structure: in Section II, we review the RA algorithm. We also discuss in that section a previous result from [32]-[33] that shows that the RA emphasis function is composed of two well-differentiated terms, allowing the introduction of a new emphasis criterion where those two factors are mixed in a flexible manner. Then, Section III-A presents the main contribution of the paper, a modified version of RA that combines the mixed emphasis function with a method to dynamically select the parameter that weighs both emphasis factors. In Section III-B we generalize some properties of the RA algorithm to our approach, showing, from an analytical point of view, that it can be expected that our approach presents a better performance than RA. In Section IV we test the performance of our approach in a benchmark with different binary classification problems, analyzing error and convergence rates, generalization capability, and robustness with respect to parameter selection. Finally, in Section V, conclusions and future research lines are presented.

II. DEALING WITH REAL ADABOOST EMPHASIS

A. The Real Adaboost algorithm

Let us consider a set of patterns and their corresponding labels, $\{\mathbf{x}_l, d_l\}_{l=1}^L$, where $\mathbf{x}_l \in \mathcal{X}$ and $d_l \in \{-1, 1\}$, and the objective is to build a function which is able to classify new patterns as accurately as possible.

The fundamental idea of the Real Adaboost (RA) algorithm is to combine several base learners to obtain a classifier with improved performance. Concretely, to build up an RA classifier [8], at each round $t = 1, \dots, T$, a new base learner is trained with an emphasized population and is added to the ensemble. The final ensemble output, $f_T(\mathbf{x})$, is given by a weighted linear combination of the outputs of all the

learners, so that the estimated label is

$$\hat{d}(\mathbf{x}) = \text{sign}[f_T(\mathbf{x})] = \text{sign}\left[\sum_{t=1}^T \alpha_t o_t(\mathbf{x})\right] \quad (1)$$

where α_t is the output weight assigned to the t -th base learner, and $o_t(\mathbf{x}) : \mathcal{X} \rightarrow [-1, 1]$ is the function implemented by such learner to minimize a weighted quadratic error, namely

$$E_t = \sum_{l=1}^L D_t(\mathbf{x}_l) [d_l - o_t(\mathbf{x}_l)]^2 \quad (2)$$

$D_t(\mathbf{x})$ being the emphasis function of the t -th learner, which indicates how much attention must be paid to each training sample. Initially, all the training samples have the same importance, i.e., $D_1(\mathbf{x}_l) = 1/L$, $\forall l = 1, \dots, L$, and the emphasis function is updated at each round according to

$$D_{t+1}(\mathbf{x}_l) = \frac{D_t(\mathbf{x}_l) \exp[-\alpha_t o_t(\mathbf{x}_l) d_l]}{Z_t} \quad (3)$$

where Z_t is a normalization factor that assures $\sum_{l=1}^L D_{t+1}(\mathbf{x}_l) = 1$.

Regarding the output weight α_t , RA goal is to calculate it so that the following bound on the ensemble training error is minimized at each round:

$$B_t = \frac{1}{L} \sum_{l=1}^L \exp[-f_t(\mathbf{x}_l) d_l] \geq \frac{1}{2L} \sum_{l=1}^L |\text{sign}[f_t(\mathbf{x}_l)] - d_l| = \text{Training Error} \quad (4)$$

As it is shown in [8], the indirect minimization of B_t by means of minimizing the looser bound

$$B_t \leq \frac{1}{L} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l) d_l] \left[\frac{1 + o_t(\mathbf{x}_l) d_l}{2} \exp(-\alpha_t) + \frac{1 - o_t(\mathbf{x}_l) d_l}{2} \exp(\alpha_t) \right] \quad (5)$$

leads to

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 + \gamma_t}{1 - \gamma_t} \right) \quad (6)$$

where γ_t , usually referred as the edge of the t -th learner,

$$\gamma_t = \sum_{l=1}^L D_t(\mathbf{x}_l) o_t(\mathbf{x}_l) d_l \quad (7)$$

indicates the quality of each weak learner by measuring the correlation of the learner outputs and the labels (weighted with the emphasis function).

In theory, the RA algorithm can be used to boost the performance of any base learner which is only slightly better than random to get an ensemble of arbitrarily high accuracy [34]. Therefore, it is common in the related literature to refer to the base learners also as weak learners. In the practice, however, after a few rounds of the algorithm it might not be easy to build better-than-random rules, and both the strength of the learners and an adequate selection of their design parameters play a significant role in the performance of the overall ensemble [35], [36], thus justifying the use of stronger learners, such as neural networks, as the base components of RA ensembles (see, among others, [37], [38]).

B. Emphasizing critical and erroneous samples

In principle, emphasis function (3) was proposed to pay more attention to the most erroneous samples during RA training. However, if this emphasis function is analyzed in more detail [32], it can be shown that it is really composed of two well-differentiated terms which are combined in a fixed way. To see this, let us first define the partial output of the RA ensemble up to time t as

$$f_t(\mathbf{x}_l) = \sum_{t'=1}^t \alpha_{t'} o_{t'}(\mathbf{x}_l) \quad (8)$$

Using this expression, we can rewrite (3) as

$$\begin{aligned} D_{t+1}(\mathbf{x}_l) &= \frac{D_t(\mathbf{x}_l) \exp[-\alpha_t o_t(\mathbf{x}_l) d_l]}{Z_t} = \frac{\prod_{t'=1}^t \exp[-\alpha_{t'} o_{t'}(\mathbf{x}_l) d_l]}{L \prod_{t'=1}^t Z_{t'}} \\ &= \frac{\exp[\sum_{t'=1}^t -\alpha_{t'} o_{t'}(\mathbf{x}_l) d_l]}{L \prod_{t'=1}^t Z_{t'}} = \frac{\exp[-f_t(\mathbf{x}_l) d_l]}{L \prod_{t'=1}^t Z_{t'}} \end{aligned} \quad (9)$$

Now, taking into account that $-2f_t(\mathbf{x}_l) d_l = [f_t(\mathbf{x}_l) - d_l]^2 - f_t^2(\mathbf{x}_l) - d_l^2$, and that $d_l^2 = 1$, (9) can alternatively be expressed as

$$D_{t+1}(\mathbf{x}_l) = \frac{1}{\tilde{Z}_t} \exp \left[\frac{[f_t(\mathbf{x}_l) - d_l]^2}{2} \right] \exp \left[-\frac{f_t^2(\mathbf{x}_l)}{2} \right] \quad (10)$$

where all constant terms have been incorporated to \tilde{Z}_t . This expression shows that the RA emphasis function consists of two exponential factors, and that each of them stands for a different kind of emphasis. The first factor pays attention to the quadratic error of each sample only, while the second focuses on the “critical” samples, i.e., those samples close to the classification boundary achieved by the ensemble output at round t ($f_t(\mathbf{x}) = 0$).

The above emphasis terms combination made us wonder if the fixed RA emphasis criterion is always the best option, leading us to propose in [33] a new emphasis function that incorporated an additional degree of flexibility to the fixed RA emphasis function:

$$D_{\lambda,t+1}(\mathbf{x}_l) = \frac{1}{Z_{\lambda,t}} \exp \{ \lambda [f_t(\mathbf{x}_l) - d_l]^2 - (1 - \lambda) f_t^2(\mathbf{x}_l) \} \quad (11)$$

where $Z_{\lambda,t}$ assures $\sum_{l=1}^L D_{\lambda,t+1}(\mathbf{x}_l) = 1$, and λ ($0 \leq \lambda \leq 1$) is a weighting parameter that trades-off the attention that the next learner should pay to the samples that are close to the boundary and to those that are classified with a large quadratic error. For instance, when $\lambda = 0$ only the critical samples are emphasized, whereas $\lambda = 1$ implies that the ensemble only emphasizes those samples with largest quadratic error. The original RA emphasis (3) is recovered when setting $\lambda = 0.5$.

Regarding the training of the ensemble weak learners when the new emphasis function is applied, (2) can still be used incorporating the new λ -dependent emphasis:

$$E_{\lambda,t} = \sum_{l=1}^L D_{\lambda,t}(\mathbf{x}_l) [d_l - o_t(\mathbf{x}_l)]^2 \quad (12)$$

Finally, as in the traditional RA approach, we obtain the output of the ensemble as the weighted linear combination of the learners outputs, i.e.

$$f_T(\mathbf{x}) = \sum_{t=1}^T \beta_t o_t(\mathbf{x}) \quad (13)$$

where the output weights have now been denoted with β_t to distinguish them from RA output weights.

The selection of the weight associated to each classifier can still be done with the aim of minimizing

the bound on the training error given by (5) (replacing α_t with β_t), using a procedure similar to that employed by the standard RA algorithm (see Appendix I):

$$\beta_t = \frac{1}{2} \ln \left(\frac{1 + \delta_t}{1 - \delta_t} \right) \quad (14)$$

where we have defined δ_t as

$$\delta_t = \frac{1}{LB_{t-1}} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l)d_l] o_t(\mathbf{x}_l)d_l \quad (15)$$

This expression is equivalent to that for γ_t (Eq. (7)), since for the RA algorithm, $D_t(\mathbf{x}_l) \propto \exp[-f_t(\mathbf{x}_l)d_l]$; however, there is a clear difference in their interpretations:

- δ_t measures the correlation between the learner outputs and the true labels, weighted by the contribution of each sample to B_{t-1} (see Eq. (4)).
- γ_t also measures the above correlation; but, in this case, each correlation term is weighted by the relevance (emphasis) of the associated data.

Obviously, both points of view are the same only if $D_{\lambda,t}(\mathbf{x}_l) = D_t(\mathbf{x}_l)$, i.e, if $\lambda = 0.5$. Therefore, when the mixed emphasis function is employed, we prefer to use definition (15), which has a clear interpretation for any λ ; it is in this sense that we refer to δ_t as a ‘‘generalized edge’’.

The application of the emphasis function (11) for a fixed value of λ , together with (12)-(15), results in a new algorithm for building boosting ensembles, to which we will refer in the sequel as RA with weighted emphasis (RA-we). This algorithm has already been applied to the construction of Multi Layer Perceptron (MLP) and Radial Basis Function Network (RBFN) based ensembles, obtaining some advantages over RA in terms of recognition accuracy and rate of convergence (see [31] and [39], respectively); obviously, to get the best performance from RA-we, the optimal value of λ has to be selected. However, this is not an easy task because this value depends on each particular classification problem. Although it is possible to select λ using cross-validation (CV), this strategy presents a limited ability to get the best from the RA-we approach [31]. In the following section we propose a dynamic method to select mixing coefficient λ ,

choosing at each round the most appropriate mixture of boundary proximity and error emphasis; therefore, modifying the value of this parameter during the growth of the ensemble.

III. BOOSTING BY DYNAMICALLY ADJUSTED WEIGHTED EMPHASIS

A. Dynamic selection of the mixing coefficient: The DW-RA algorithm

The RA-we algorithm presented in the previous section assumes that the same value λ is used to update the emphasis function at each round. In this subsection, we present a novel method for the dynamic selection of the mixing parameter that not only solves the problem of selecting an appropriate value for λ without recurring to CV, but also allows to select a different value at each round. Note that, as we can see in (11), to update the emphasis at round $t + 1$ we only need to know the partial output of the ensemble at the previous round, so there is no reason for not allowing different λ values during the construction of the ensemble.

To select the mixing coefficient at each round, we propose to consider the effect of the new learner being added, for different values of λ , on the performance of the ensemble. To be more specific, at round t we will train a set of base learners, $\{o_t^{(j)}(\mathbf{x})\}_{j=1}^J$, using populations emphasized according to (11) for a predefined set $\{\lambda^{(j)}\}_{j=1}^J$ of possible mixing parameter values. We will add to the ensemble the learner giving the largest generalized edge (15); i.e., we add $o_t^{(j^*)}(\mathbf{x})$ where

$$j^* = \arg \max_j \delta_t^{(j)}$$

We can find some justification for this selection criterion by taking the gradient of B_t with respect to weights $\beta_t^{(j)}$, $j = 1, \dots, J$, and computing its value before the new learner is added (i.e., at the instant at which $\beta_t^{(j)} = 0$),

$$\left. \frac{\partial B_t}{\partial \beta_t^{(j)}} \right|_{\beta_t^{(j)}=0} = -\frac{1}{L} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l) d_l] o_t(\mathbf{x}_l)^{(j)} d_l = -B_{t-1} \delta_t^{(j)}$$

Then, we can see that the learner that achieves the largest absolute component in the direction of the gradient of B_t is that with the largest generalized edge; this is an application of the Gauss-Southwell method [40], and it makes us expect a larger reduction in the value of B_t when adding the learner with largest δ_t .

For completeness, let us mention that the selection of the best mixing parameter at each round is not necessarily optimal from the perspective of the whole ensemble design. Nevertheless, it is our experience that this method results in a better performance than other selection criteria (for instance, selecting the median at each round, that is a classical choice for fusion applications) and, in any case, it provides very satisfactory performance in terms of recognition accuracy, as we will show in the experiments section.

Table I summarizes the main steps of the proposed Dynamically adapted Weighted emphasis version of Real Adaboost (DW-RA in the sequel).

B. Some remarks on the proposed algorithm

The good performance of the RA algorithm has been checked experimentally in many applications, and has also been justified theoretically (a summary of several RA analyses can be found in [41]). Since RA-we and DW-RA output weights are fixed in a very similar manner to that of standard RA, most of these relevant properties can be straightforwardly extended to these new algorithms. In this section, we will analyze the implications of weighted emphasis function (11). It will turn out that our criterion for dynamically selecting λ is very convenient from different perspectives.

The first two items in this subsection extend some properties of the RA algorithm to the case in which weighted emphasis (11) is used, and offer some insights into the potential advantages of DW-RA in terms of training error convergence and generalization capabilities. The third item introduces an alternative interpretation of the way in which (11) emphasizes the population, providing a new point of view about the connections between RA, RA-we, and DW-RA.

TABLE I

DW-RA ALGORITHM DESCRIPTION

1 - Inputs: $\{\mathbf{x}_l, d_l\}_{l=1}^L, \{\lambda^{(j)}\}_{j=1}^J$
2 - Train a first weak learner, $o_1(\mathbf{x})$, according to the cost function:
$E_1 = \frac{1}{L} \sum_{l=1}^L [d_l - o_1(\mathbf{x}_l)]^2$
3 - For $t = 2, \dots, T$
3.1 - For $j = 1, \dots, J$
3.1.1 - Calculate the emphasis function:
$D_{\lambda^{(j)}, t}(\mathbf{x}_l) = \frac{1}{Z_t} \exp \{ \lambda^{(j)} [f_{t-1}(\mathbf{x}_l) - d_l]^2 - (1 - \lambda^{(j)}) f_{t-1}^2(\mathbf{x}_l) \}$
3.1.2 - Train $o_t^{(j)}(\mathbf{x}_l)$ to minimize the cost function:
$E_t^{(j)} = \sum_{l=1}^L D_{\lambda^{(j)}, t}(\mathbf{x}_l) [d_l - o_t^{(j)}(\mathbf{x}_l)]^2$
3.1.3 - Calculate the new edge associated to the learner
$\delta_t^{(j)} = \frac{1}{LB_{t-1}} \sum_{l=1}^L o_t^{(j)}(\mathbf{x}_l) d_l \exp[-f_{t-1}(\mathbf{x}_l) d_l]$
3.2 - Add to the ensemble:
$o_t(\mathbf{x}) = o_t^{(j^*)}(\mathbf{x}) \text{ where } j^* = \arg \max_j \delta_t^{(j)}$
3.3 - Calculate the output weight for the new weak learner
$\beta_t = \frac{1}{2} \ln \left(\frac{1 + \delta_t}{1 - \delta_t} \right)$
where $\delta_t = \delta_t^{(j^*)}$
4 - Output the final classifier:
$f_T(\mathbf{x}) = \sum_{t=1}^T \beta_t o_t(\mathbf{x})$

1) *Training error convergence:* to analyze the training error behavior of RA-we and DW-RA, let us begin from a result presented by Schapire and Singer in [8]. Recall that B_t was defined as an upper bound for the training error, and that RA was in fact minimizing the upper bound for B_t given by (5). Then, according to [8], when the output weights of the ensemble are selected with (6)-(7), the bound in (5) can be rewritten as

$$B_t \leq \prod_{t'=1}^t \sqrt{1 - \gamma_{t'}^2} \quad (16)$$

Defining the square of the overall edge of the ensemble as $\gamma^2 = \min_{t'=1, \dots, t} \{\gamma_{t'}^2\}$ and taking into

account that $1 - x \leq \exp(-x)$ for $x > 0$, it also holds that

$$B_t \leq (1 - \gamma^2)^{\frac{t}{2}} \leq \exp\left[-\frac{\gamma^2}{2}t\right] \quad (17)$$

Given that B_t is itself an upper bound for the training error, this result can be used as an indication that the training error of RA decreases approximately exponentially with the number of rounds [8].

As we explained in Subsection II-B, our criterion for fixing β_t minimizes exactly the same bound for B_t (see also Appendix I). Therefore, a similar expression is found for RA-we and DW-RA if the edge values are obtained according to (15) (see Appendix II for a demonstration):

$$B_t \leq \prod_{t'=1}^t \sqrt{1 - \delta_{t'}^2} \quad (18)$$

from which, defining $\delta^2 = \min_{t'=1, \dots, t} \{\delta_{t'}^2\}$,

$$B_t \leq (1 - \delta^2)^{\frac{t}{2}} \leq \exp\left[-\frac{\delta^2}{2}t\right] \quad (19)$$

This result shows that the training error of RA-we and DW-RA will also decrease approximately exponentially with the number of rounds, and, what is more important to our discussion here, it explains why one could expect a faster reduction of DW-RA training error than from any of the other two methods. Effectively, we know that the output weight selection when the mixed emphasis function is used (which includes $\lambda = 0.5$ as a special case) results, at each round, in bound (18). Now, note that DW-RA trains at each round a pool of candidate learners for different values of the the mixing parameter, choosing the one which gets a largest δ_t , while RA and RA-we are limited to only one learner with a predetermined λ . This way, it seems plausible that DW-RA gets a faster reduction of (18) and, since B_t is itself an upper bound on the training error, a faster reduction of the training error. However, this will not always be the case, since optimality at each round does not guarantee optimality from the perspective of the whole ensemble. In the experimental section we will give further evidence about the ability of DW-RA to provoke a fast convergence of the ensemble error.

2) *Generalization properties:* let us assume that the data of a given classification problem are drawn independently from distribution P ; then, the generalization capabilities of the designed network can be expressed in terms of the *expected risk*

$$R[f_T] = E_{(\mathbf{x}, d) \sim P} \left\{ \frac{1}{2} | \text{sign} [f_t(\mathbf{x})] - d | \right\}$$

According to [8], for the RA algorithm $R[f_T]$ can be upper bounded with probability at least $1 - \epsilon$ ($\epsilon > 0$)

by

$$R[f_T] \leq R_T^{\text{margin}}(\theta) + O \left(\frac{1}{\sqrt{L}} \sqrt{\frac{\vartheta \log^2 L / \vartheta}{\theta^2} + \log \frac{1}{\epsilon}} \right) \quad (20)$$

where L is the size of the training data set, ϑ is the VC-dimension of the space of the functions implemented by the weak learners [42], and $R_T^{\text{margin}}(\theta)$ is the fraction of training patterns with a classification margin

$$\rho_T(\mathbf{x}_l) = -\frac{f_T(\mathbf{x}_l)d_l}{\sum_{t=1}^T \alpha_t} \quad (21)$$

smaller than or equal to some $\theta \in (0, 1]$.

The above bound for the generalization risk is valid for any ensemble that calculates its global output as a linear combination of the outputs of several learners, including also RA-we and DW-RA (considering that in these cases the margin is normalized by $\{\beta_t\}$ values, i.e., $\rho_T(\mathbf{x}_l) = -f_T(\mathbf{x}_l)d_l / \sum_{t=1}^T \beta_t$). Furthermore, it can be seen that, for a given classification problem and weak learner complexity, the second term in (20) has the same value for RA, RA-we and DW-RA, so that the generalization bound is only dependent on the margin risk $R_T^{\text{margin}}(\theta)$. Therefore, to get a better understanding about the generalization of the three algorithms, it is interesting to spend some effort in the study of $R_T^{\text{margin}}(\theta)$ (a more detailed analysis about the margin risk can be found in [43]). There are two results that directly connect the classification margin, ρ_T , to the edge of the weak learners.

- a) The first result, applied to RA-we and DW-RA, is obtained from the analysis of the training error bound B_t , since, at the t -th round, it can be expressed in terms of the classification margin ρ_t as

follows:

$$B_t = \frac{1}{L} \sum_{l=1}^L \exp[-f_t(\mathbf{x}_l)d_l] = \frac{1}{L} \sum_{l=1}^L \exp \left[-\rho_t(\mathbf{x}_l) \sum_{t'=1}^t \beta_{t'} \right] \quad (22)$$

This indicates that the minimization of B_t also results in an indirect maximization of the classification margins. Now, remember our previous discussion about the ability of DW-RA at reducing B_t ; then, we can also expect that DW-RA is more effective than both RA-we and RA at maximizing the margin of classification and, therefore, at minimizing the expected risk of the ensemble.

- b) The second result is the Min-Max Theorem [6], that establishes a connection between the maximum value that the ensemble margin can achieve and the smallest of the weak learner edges by applying the duality theorem for Linear Programs (LPs) [44] to boosting methods. To introduce this theorem for the RA algorithm, let us firstly define the classification margin as a function of the weights of the ensemble

$$\rho(\boldsymbol{\alpha}) = \min_{l=1, \dots, L} \rho_t(\mathbf{x}_l) = \min_{l=1, \dots, L} -\frac{f_T(\mathbf{x}_l)d_l}{\sum_{t=1}^T \alpha_t} \quad (23)$$

where $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_T]^T$. Now, defining the learner edge as function of the probability vector $\mathbf{D}_t = [D_t(\mathbf{x}_1), \dots, D_t(\mathbf{x}_L)]^T$

$$\gamma(\mathbf{D}_t) = \max_{t=1, \dots, T} \gamma_t(\mathbf{D}_t) = \max_{t=1, \dots, T} \sum_{l=1}^L D_t(\mathbf{x}_l) o_t(x_l) d_l \quad (24)$$

then, the Min-Max Theorem establishes that

$$\rho_{\max} = \max_{\boldsymbol{\alpha}} \rho(\boldsymbol{\alpha}) = \min_{\mathbf{D}_t} \gamma(\mathbf{D}_t) = \gamma_{\min} \quad (25)$$

what indicates that the maximal achievable margin, ρ_{\max} , equals the minimal edge, γ_{\min} .

To apply this theorem to RA-we and DW-RA, we must define the classification margin as a function of values β_t ,

$$\rho(\boldsymbol{\beta}) = \min_{l=1, \dots, L} \rho_t(\mathbf{x}_l) = \min_{l=1, \dots, L} -\frac{f_T(\mathbf{x}_l)d_l}{\sum_{t=1}^T \beta_t} \quad (26)$$

where $\boldsymbol{\beta} = [\beta_1, \dots, \beta_T]^T$, and we must also change $\gamma(\mathbf{D}_t)$ by the generalized edge expression, making it dependent from the previous global output instead of the emphasis function; i.e., let us define

$$\delta(\mathbf{f}_{t-1}) = \max_{t=1, \dots, T} \delta_t(\mathbf{f}_{t-1}) = \max_{t=1, \dots, T} \frac{1}{LB_{t-1}} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l)d_l] o_t(\mathbf{x}_l)d_l \quad (27)$$

where $\mathbf{f}_{t-1} = [f_{t-1}(\mathbf{x}_1), \dots, f_{t-1}(\mathbf{x}_L)]^T$; then, we can reformulate the Min-Max Theorem in the form

$$\rho_{\max} = \max_{\boldsymbol{\beta}} \rho(\boldsymbol{\beta}) = \min_{\mathbf{f}_{t-1}} \delta(\mathbf{f}_{t-1}) = \delta_{\min} \quad (28)$$

This theorem connects the maximum value that ensemble margin can achieve, ρ_{\max} , with the minimum edge that is obtained through the optimization of the weak learners, δ_{\min} . As we have already explained, we can expect that in general DW-RA will present a larger minimum edge and, therefore, it will usually achieve a lower $R_T^{\text{margin}}(\theta)$, and thus a better generalization performance than RA and RA-we.

3) *Weak learners training:* the above remarks show that maximizing the edge of the new learners to be incorporated to the ensemble is critical to optimize the overall performance. Then, one may wonder why do we not train the weak learners so that this value is maximized directly. This idea is not new and several authors [8], [41] have already proposed to train the RA weak learners maximizing directly the learner edge γ_t , i.e., maximizing the following cost function (see (7)):

$$E_t^\gamma = \sum_{l=1}^L D_t(\mathbf{x}_l) o_t(\mathbf{x}_l) d_l \quad (29)$$

However, in most cases the maximization of (29) is a difficult task, since the learner edge is not a convex function of the learner weights. Thus, as we have discussed in Subsection II-A, it is usually easier to train each weak learner so that the average quadratic error, weighted by emphasis function D_t , is minimized.

At first glance, the weighted quadratic error does not show any relation with γ_t , but analyzing it in more detail, it can be reformulated as a regularized version of (29):

$$\begin{aligned} E_t &= \sum_{l=1}^L D_t(\mathbf{x}_l) [d_l - o_t(\mathbf{x}_l)]^2 = \sum_{l=1}^L D_t(\mathbf{x}_l) [d_l^2 + o_t^2(\mathbf{x}_l) - 2o_t(\mathbf{x}_l)d_l] \\ &= \sum_{l=1}^L D_t(\mathbf{x}_l)d_l^2 + \sum_{l=1}^L D_t(\mathbf{x}_l)o_t^2(\mathbf{x}_l) - 2 \sum_{l=1}^L D_t(\mathbf{x}_l)o_t(\mathbf{x}_l)d_l \end{aligned} \quad (30)$$

and neglecting the first term because of $\sum_{l=1}^L D_t(\mathbf{x}_l)d_l^2 = 1$, we arrive to

$$E_t \stackrel{c}{=} \sum_{l=1}^L D_t(\mathbf{x}_l)o_t^2(\mathbf{x}_l) - 2 \sum_{l=1}^L D_t(\mathbf{x}_l)o_t(\mathbf{x}_l)d_l \quad (31)$$

where ' $a \stackrel{c}{=} b$ ' is used to denote that a and b only differ in a constant. The first term in the right hand side of (31) is a sort of regularizer, since it penalizes large outputs, while minimizing the second term, is equivalent to maximizing E_t^γ .

When the training samples are emphasized according to the mixed emphasis function (11), the relation between cost function $E_{\lambda,t}$ and the edge is given by

$$E_{\lambda,t} \stackrel{c}{=} \sum_{l=1}^L D_{\lambda,t}(\mathbf{x}_l)o_t^2(\mathbf{x}_l) - 2E_{\lambda,t}^\delta \quad (32)$$

where, as we show in Appendix III,

$$E_{\lambda,t}^\delta = \frac{\tilde{Z}}{LB_{t-1}} \sum_{l=1}^L G_{\lambda,t}(\mathbf{x}_l) \exp[-f_{t-1}(\mathbf{x}_l)d_l] o_t(\mathbf{x}_l)d_l \quad (33)$$

\tilde{Z} being a constant, irrelevant for minimization purposes, and $G_{\lambda,t}(\mathbf{x}_l)$ being a new emphasis function defined as

$$G_{\lambda,t}(\mathbf{x}_l) = \frac{1}{Z_{G,t}} \exp \left\{ 2(\lambda - 0.5) \left[f_{t-1}(\mathbf{x}_l) - \frac{d_l}{2} \right]^2 \right\} \quad (34)$$

where $Z_{G,t}$ is a normalization factor to get $\sum_{l=1}^L G_{\lambda,t}(\mathbf{x}_l) = 1$.

Comparing (33) and (15), it is possible to conclude that the new weak learners incorporated to the RA-we and DW-RA ensembles are trained to maximize an emphasized (and regularized) version of the learner edge, δ_t , with different values of λ standing for different kinds of emphasis; for instance:

i) If $\lambda = 0$ (only boundary emphasis is employed), we get

$$G_{\lambda,t}(\mathbf{x}_l) \Big|_{\lambda=0} = \frac{1}{Z_{G,t}} \exp \left\{ - \left[f_{t-1}(\mathbf{x}_l) - \frac{d_l}{2} \right]^2 \right\} \quad (35)$$

and then the samples which are correctly classified with an output value close to 0.5 or -0.5 are considered the most relevant ones in the maximization of δ_t .

ii) When classical RA is used, $\lambda = 0.5$, $G_{\lambda,t}(\mathbf{x}_l)$ is a uniform distribution, thus all the samples have the same influence on the edge maximization.

iii) Finally, if $\lambda = 1$

$$G_{\lambda,t}(\mathbf{x}_l) \Big|_{\lambda=1} = \frac{1}{Z_{G,t}} \exp \left\{ \left[f_{t-1}(\mathbf{x}_l) - \frac{d_l}{2} \right]^2 \right\} \quad (36)$$

and the samples with the largest quadratic error are the most emphasized in the maximization of the edge.

Therefore, whereas RA considers that all the samples have the same influence in the maximization of the edge, RA-we allows us to emphasize differently the influence of the samples, and the DW-RA algorithm permits to modify this influence along the ensemble growing. As already discussed, this additional degree of freedom can be exploited to improve the performance of boosting schemes. In the next section, we show that this is indeed the case when selecting λ by means of the DW-RA algorithm.

IV. EVALUATION OF THE PROPOSED ALGORITHM

To illustrate the advantages of DW-RA, in this section we compare its performance with that of the RA algorithm, as well as with that of RA-we when λ is selected by a cross-validation (CV) process.

A. Data sets and settings description

The performance comparisons have been performed on eight binary problems: *Kwok* and *Ripley* (two synthetic problems from [45] and [46], respectively); *Abalone*, *Image*, *Contraceptive*, *Spam* and *Tictactoe*

TABLE II
CHARACTERISTICS OF THE BENCHMARK PROBLEMS

Problem	Notation	# <i>dim</i>	Train samples (n_1/n_{-1})	Test samples (n_1/n_{-1})	SVM error rate
<i>Abalone</i>	<i>Ab</i>	8	1238/1269	843/827	20.9
<i>Contraceptive</i>	<i>Co</i>	9	506/377	338/252	28.61
<i>Image</i>	<i>Im</i>	18	821/1027	169/293	2.46
<i>Kwok</i>	<i>Kw</i>	2	300/200	6120/4080	11.74
<i>Phoneme</i>	<i>Ph</i>	5	952/2291	634/1527	15.35
<i>Ripley</i>	<i>Ri</i>	2	125/125	500/500	9.8
<i>Spam</i>	<i>Sp</i>	57	1673/1088	1115/725	7.2
<i>Tictactoe</i>	<i>Ti</i>	9	199/376	133/250	1.7

(five real datasets from the UCI repository [47]); and *Phoneme* (available at [48]). *Abalone* is originally a multiclass classification problem, converted into binary according to [49]. The last four problems of this collection do not have a predefined train/test partition, hence each data set has been randomly divided into 10 subsets of (approximately) equal size, and each time 6 of the subsets are selected to train the ensemble and the remaining 4 subsets are used to test the performance of the classifiers. In Table II we offer a summary of the main characteristics of these data sets: we give the simplified notation that will be used in the tables, the number of dimensions (*#dim*) and the number of samples belonging to each class (n_1/n_{-1}) in both the training and test sets; finally, we have also added a column with the results achieved by a Support Vector Machine (SVM) with Gaussian Kernel to provide a reference result from a state-of-the-art classification technique. To design these SVMs, we have used the IRWLS SVM toolbox for Matlab, available at [50], optimizing the kernel dispersion and penalty factor with a five fold CV process.

As we have already mentioned, three kinds of ensembles, according to three different boosting me-

thods, have been considered:

- The conventional RA algorithm described in Section II-A; to clarify the notation, we will denote it as RA-se (RA with standard emphasis).
- An ensemble using weighted emphasis function (11), with a fixed mixing coefficient selected from $\{0, 0.1, 0.2, \dots, 1\}$ using a five fold CV procedure with 20 iterations over each partition. To simplify the notation, this method will be denoted as CV RA-we.
- The proposed DW-RA algorithm selecting λ (at each iteration) from within the same 11 values explored for the previous method.

To select the final number of rounds T for each ensemble, we have paid attention to the evolution of α_t , because when these values become very close to zero the influence of new learners in the ensemble performance is very reduced; concretely, the ensemble growth is stopped when the ratio between the average α_t in the last 10 rounds and the sum of all α_t is lower than a stop value (C_{stop}), i.e.,

$$\frac{\sum_{t'=T-9}^T \alpha_{t'}}{10 \sum_{t'=1}^T \alpha_{t'}} < C_{\text{stop}} \quad (37)$$

where C_{stop} has been experimentally fixed to 0.01 for all the algorithms under consideration².

To build up the ensembles, we have used Multi Layer Perceptrons (MLPs) as base learners, with one hidden layer composed of M hidden neurons, and hyperbolic tangent activations for both the output and the hidden neurons. Initially, all the weights are set to random values in interval $[-0.1, 0.1]$, and then a back-propagation algorithm is used to minimize cost function (2) in RA-se, or (12) for CV RA-we and DW-RA; the back-propagation learning step is decreased linearly from $\mu = 0.01$ to 0 along 100 epochs, which we have checked that are enough to assure the convergence of the networks. To avoid overfitting, 20% of the data used to train the base learner is set aside for validation, stopping the training if necessary.

²In *Tictactoe* we have observed that the algorithms convergence is slower; so, in this case, it was necessary to fix $C_{\text{stop}} = 5 \cdot 10^{-6}$ for RA-se, and $C_{\text{stop}} = 10^{-5}$ for CV RA-we and DW-RA.

For each of these algorithms, the number of MLP hidden units (M) has been selected from the set $\{2, 3, \dots, M_{\max}\}$ by a five fold CV process, with 20 iterations over each partition, that selects the optimum M for the ensemble. The explored range varies from 2 to the value that guarantees that there are at least 4 training samples for each MLP parameter (we have checked that larger M values cause performance degradations except for *Ripley*, where we have relaxed this condition and explored until $M = 60$).

To evaluate the performance of the different algorithms (RA-se, CV RA-we and DW-RA), we have obtained the test Classification Error rates (CE) for 50 runs, each one using independent initializations in all the learners that compose the ensemble, as well as a different random order of the data for each backpropagation training. In the tables we will present, for each algorithm and problem, the estimated mean and standard deviation of the CE (calculated over these 50 runs).

B. Performance analysis with base learners of fixed complexity

In this subsection we analyze the performance of the three ensembles for a fixed base learner complexity (i.e., fixed M). To do so, we have first used CV to select the best M for the RA-se ensemble, and then, we have carried out experiments for the other two algorithms using the same M . In this way, we are isolating the effect of applying the mixed emphasis function instead of the standard one. However, note that, in principle, this comparison benefits RA-se, given that the complexity of the base learners could also be optimized for the CV RA-we and DW-RA ensembles; this issue will be analyzed in the next subsection.

Table III summarizes the average CE achieved by each method, indicating with boldface the best result for each problem. The table shows also the value of M selected for each problem, the size of the ensembles and the value of the mixing parameter selected for the CV RA-we algorithm (λ_{CV}).

In the light of these results, we can conclude that both CV RA-we and DW-RA generally achieve lower classification rates than conventional RA-se; this confirms the usefulness of the mixed emphasis

TABLE III

CLASSIFICATION ERROR RATES (CE) AND ENSEMBLE SIZES (T) ACHIEVED BY RA-se, CV RA-we AND DW-RA WHEN THE NUMBER OF HIDDEN UNITS OF THE MLPs (M) IS OPTIMIZED FOR THE RA-se ALGORITHM

	M	RA-se		λ_{CV}	CV RA-we		DW-RA	
		T	CE (%)		T	CE (%)	T	CE (%)
<i>Ab</i>	4	31.18 (± 2.55)	19.38 (± 0.15)	0.8	33.06 (± 4.17)	19.45 (± 0.17)	37.45 (± 2.26)	19.09 (± 0.14)
<i>Co</i>	2	33.68 (± 5.22)	29.00 (± 1.45)	0.1	24.94 (± 2.76)	28.85 (± 1.48)	45.80 (± 4.31)	28.54 (± 1.25)
<i>Im</i>	11	19.60 (± 2.69)	2.46 (± 0.31)	0.5	19.60 (± 2.69)	2.46 (± 0.31)	29.74 (± 3.39)	2.35 (± 0.28)
<i>Kw</i>	15	29.26 (± 0.99)	11.71 (± 0.05)	0.5	29.26 (± 0.99)	11.71 (± 0.05)	32.46 (± 1.62)	11.70 (± 0.05)
<i>Ph</i>	60	27.74 (± 2.40)	14.04 (± 0.52)	0.0	16.98 (± 0.99)	13.73 (± 0.57)	37.63 (± 4.81)	13.45 (± 0.57)
<i>Ri</i>	48	28.86 (± 1.28)	9.73 (± 0.09)	0.7	36.72 (± 4.38)	9.58 (± 0.21)	36.60 (± 3.09)	9.41 (± 0.18)
<i>Sp</i>	7	26.18 (± 4.50)	5.94 (± 0.61)	0.5	26.18 (± 4.50)	5.94 (± 0.61)	35.74 (± 4.52)	5.75 (± 0.50)
<i>Ti</i>	4	5631.78 (± 1105.42)	0.75 (± 0.55)	0.5	5631.78 (± 1105.42)	0.75 (± 0.55)	5698.42 (± 1044.33)	0.79 (± 0.55)

function for constructing ensembles of improved performance. We can see that CV RA-we selects a mixing parameter different from 0.5 only in four out of the eight problems, while for the other four it improves RA-se performance in three cases.

When comparing DW-RA to RA-se, results are even more encouraging: DW-RA achieves better performance in six of the benchmark problems: *Abalone*, *Contraceptive*, *Image*, *Phoneme*, *Ripley* and *Spam*. Comparing the means and standard deviations of the CE s, we can conclude that this difference in behavior is normally quite clear, and only in some cases (*Contraceptive* and *Image*) the large standard deviations, in relation to the difference between the average CE s for RA-se and DW-RA, make results less conclusive. Finally, in the *Kwok* dataset both algorithms are practically tied, while for *Tictactoe* it is

RA-se which gets a better performance but, again, results show large standard deviations in comparison to the difference between the mean CE s. It is important to remark the difficulty of achieving more important reductions of the CE due to the fact that RA-se and CV RA-we ensembles already present very good performance (see in Table II that these algorithms get similar or, in most the cases, better performance than SVM).

Looking at the final ensemble sizes, we can also notice that DW-RA builds ensembles with slightly higher complexity than either RA-se or CV RA-we. Then, one may wonder if DW-RA suffers a slower convergence, or even if the advantages of DW-RA are due to a larger number of rounds. Neither of these happen, as we can check in Fig. 1, where we have depicted the evolution of average test CE during the construction of the ensembles for all the problems in the benchmark. We can see that, for a fixed number of rounds, DW-RA normally provides better results than any of the other two methods, and that its convergence is usually faster.

As a summary, the previous results not only allow us to conclude that the mixed emphasis function is beneficial for building ensembles, but also show that the dynamic management of the mixing parameter is crucial if one wants to get all the benefit from the idea.

We end this subsection by giving some empirical evidence related to the theoretical results presented in Subsection III-B:

- *Convergence of B_t* : Our analysis in Subsection III-B.1 predicted that DW-RA could potentially obtain a faster reduction of the bound on the training error, B_t . This result is empirically illustrated here by showing value B_t achieved by the three ensembles in several problems in the dataset (Fig. 2). Indeed, this behavior partially explains also the faster CE convergence observed in Fig. 1.
- *Margin distributions*: Although the CE rates obtained by DW-RA clearly show its good generalization capability, we have also analyzed the margin distribution $R_T^{\text{margin}}(\theta)$ achieved by the different approaches to check the theoretical analysis of Subsection III-B.2. Note that comparing $R_T^{\text{margin}}(\theta)$

for different algorithms only makes sense for a fixed base learner complexity, which is the case we are considering. We have observed that in most problems the global behaviors of the $R_T^{\text{margin}}(\theta)$ distributions are quite similar for all the algorithms, except for *Ripley* (see Fig. 3), where the risk of DW-RA is clearly lower for most values θ . Focusing on the behavior of $R_T^{\text{margin}}(\theta)$ around 0, and in particular on the minimum margin ($\rho_{\min} = \min_{l=1,\dots,L} \rho_l$), we have observed that its value is usually lower in the DW-RA algorithm than in the other approaches; for instance, in *Image* (see Fig. 4(a)) we can see that ρ_{\min} is around -0.2 for RA-se and CV RA-we, whereas DW-RA gets a value close to -0.1 ; even more, the evolution of ρ_{\min} (Fig. 4(b)) shows that DW-RA achieves lower minimum margin values from the first rounds.

C. Performance analysis with independent M selection

Rather than using the number of hidden neurons that was found optimal for RA-se, in this subsection we will cross-validate M independently for each ensemble type, which is the common procedure under which the different ensembles will be designed. The corresponding results are reported in Table IV, including also the value of M that was selected for each algorithm.

The discussion of these results can be carried out in very similar terms to those in the previous subsection. RA-se is outperformed by CV RA-we and DW-RA in most of the problems, and the latter is generally achieving smaller CE s. Comparing the results in Tables III and IV, we find that the independent cross-validation of M provides some extra improvement in terms of the recognition accuracy of the DW-RA ensemble for *Abalone*, *Image* and *Kwok* (for which, in this case, the superiority of DW-RA over RA-se is more clear).

In addition to the above discussion in terms of recognition accuracy, it is very interesting to analyze other properties of the algorithms. In particular, it is worth mentioning the following general behavior aspects, that have emerged when analyzing in more detail our experimental work:

TABLE IV

CLASSIFICATION ERROR RATES (CE) AND ENSEMBLE SIZES (T) ACHIEVED BY RA-se, CV RA-we AND DW-RA WHEN M IS SELECTED INDEPENDENTLY FOR EACH ENSEMBLE TYPE

	RA-se			CV RA-we			DW-RA			
	M	T	CE (%)	M	λ_{CV}	T	CE (%)	M	T	CE (%)
<i>Ab</i>	4	31.18 (± 2.55)	19.38 (± 0.15)	4	0.8	33.06 (± 4.17)	19.45 (± 0.17)	5	37.74 (± 1.97)	18.97 (± 0.13)
<i>Co</i>	2	33.68 (± 5.22)	29.00 (± 1.45)	6	0.1	26.04 (± 6.11)	28.60 (± 1.47)	2	45.80 (± 4.31)	28.54 (± 1.25)
<i>Im</i>	11	19.60 (± 2.69)	2.46 (± 0.31)	11	0.5	19.60 (± 2.69)	2.46 (± 0.31)	9	31.26 (± 2.65)	2.31 (± 0.29)
<i>Kw</i>	15	29.26 (± 0.99)	11.71 (± 0.05)	15	0.5	29.26 (± 0.99)	11.71 (± 0.05)	22	31.62 (± 1.99)	11.66 (± 0.05)
<i>Ph</i>	60	27.74 (± 2.40)	14.04 (± 0.52)	62	0.0	16.56 (± 0.93)	13.49 (± 0.63)	70	38.20 (± 4.91)	13.43 (± 0.63)
<i>Ri</i>	48	28.86 (± 1.28)	9.73 (± 0.09)	34	0.7	38.80 (± 7.43)	9.64 (± 0.19)	44	36.02 (± 3.09)	9.41 (± 0.18)
<i>Sp</i>	7	26.18 (± 4.50)	5.94 (± 0.61)	7	0.5	26.18 (± 4.50)	5.94 (± 0.61)	8	36.62 (± 3.92)	5.75 (± 0.51)
<i>Ti</i>	4	5631.78 (± 1105.42)	0.75 (± 0.55)	3	0.4	6965.23 (± 1141.32)	0.89 (± 0.60)	4	5698.42 (± 1044.33)	0.79 (± 0.55)

- *Size of the classifiers and error convergence:* For completeness, in Fig. 5 we show average CE evolution as a function of the number of learners for all the algorithms under consideration. Again, the slightly higher complexity observed for DW-RA ensembles is only due to DW-RA achieving a lower CE , and not to a slower convergence.
- *Stopping criterion:* Also, from Fig. 5 and Table IV, we can conclude that the stopping criterion is working reasonably well. In particular, we see that it stops the growth of the ensembles when convergence is complete, while preventing the over-fitting that is occasionally observed, so that our criterion is not benefiting any of the methods.
- *Robustness with respect to the parameters:* Another interesting aspect is the sensitivity with respect

TABLE V

CE ACHIEVED BY AN “OMNISCIENT” APPROACH THAT SELECTS THE OPTIMUM NUMBER OF HIDDEN NEURONS M_0
(AND MIXING PARAMETER λ_0 FOR RA-we)

	RA-se		RA-we			DW-RA	
	M_0	CE	λ_0	M_0	CE	M_0	CE
<i>Ab</i>	6	19.14	0.1	6	19.01	5	18.97
<i>Co</i>	4	28.88	0	3	28.51	2	28.54
<i>Im</i>	9	2.29	0.3	9	2.26	15	2.13
<i>Kw</i>	9	11.64	0.4	9	11.64	22	11.66
<i>Ph</i>	54	13.65	0.1	54	13.46	70	13.43
<i>Ri</i>	48	9.73	1	34	9.30	28	9.30
<i>Sp</i>	5	5.78	0.6	6	5.77	5	5.63
<i>Ti</i>	4	0.75	0.5	4	0.75	4	0.79

to the number of MLP hidden units. To analyze it, in Table V we show the best possible test CE that would have been achieved for each algorithm if we had validated our designs directly on the test set, as well as the optimal value M_0 (in the case of the RA-we algorithm, we have also included the best possible mixing parameter, λ_0); although this is a perverse trick for designing purposes, this “omniscient” approach is very useful to evaluate if the different approaches find successfully the appropriate values of M . Comparing these results with those in Table IV, we can conclude that DW-RA is more robust than RA-se and CV RA-we in selecting M , since in five out of the eight data sets optimum value M_0 is selected, whereas RA-se only selects the optimal value twice and CV RA-we selects it once (although optimum mixing parameter λ_0 is not selected); these cases appear with boldface numbers in Table V.

D. Additional issues

The results we have presented in this section are only a part of our extensive simulation work with the mixed emphasis function and the criterion for dynamic selection of λ . We think, however, that a discussion about CV RA-we and DW-RA performance would be incomplete without a reference to the following sets of experiments that we have also carried out:

- We have done experiments to check that selecting the mixing parameter that achieves the maximum edge is really a good choice; we have built ensembles similar to DW-RA, but selecting in each round the mixing parameter that gives the median value of the edges set $\{\delta^{(j)}\}_{j=1}^J$, a frequent strategy to construct ensembles. The *CEs* achieved by this new method not only are worst than DW-RA, but also worst than RA-se in most cases.
- The second set of experiments consisted in training RA-se networks, but selecting at each round from within 11 learners (all of them trained with $\lambda = 0.5$, but using different initializations) the one with a largest edge. Results do not differ significantly from those shown in Tables III and IV for RA-se. In this way, we could confirm that the good behavior of DW-RA was due to the use of the mixed emphasis, and not to local minima avoidance in the training of the base learners.

V. CONCLUSIONS AND FUTURE WORK

A new boosting method for building ensembles has been presented. This algorithm uses a mixed emphasis function that combines the attention to the samples error and to their proximity to the classification border, and dynamically selects the emphasis trade-off parameter that provides the learner with the largest edge, i.e., the most qualified learner at each boosting round.

A theoretical analysis shows the convenience of our proposal in terms of training error convergence and generalization capability. Experimental results corroborate this analysis, showing significant performance improvements with respect to classical Real Adaboost algorithm and to Real Adaboost with weighted

emphasis when selecting the mixing parameter by means of a direct cross-validation process.

In principle, the new algorithms can be useful to boost the performance of any kind of weak learners. However, in this paper we have focused on using MLPs as base learners, and more experimental work would be necessary to corroborate if the observed advantages apply also to other components. Furthermore, our findings suggest the interest of extending the idea of the mixed emphasis function to the construction of other kind of multi-net systems, a research line where we are currently working. Finally, we are also exploring the application of these concepts to build ensembles of ensembles.

ACKNOWLEDGMENTS

The authors would like to thank the Associate Editor and the three anonymous reviewers for their very valuable comments and suggestions.

REFERENCES

- [1] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.
- [2] A. J. C. Sharkey, ed., *Combining Artificial Neural Nets. Ensemble and Modular Multi-Net Systems*. London, UK: Springer-Verlag, 1999.
- [3] R. A. Jacobs and M. I. Jordan, "A competitive modular connectionist architecture," in *Advances in Neural Information Processing Systems 3*, D. Touretzky, ed., San Mateo, CA: Morgan Kaufmann, 1991, pp. 767–773.
- [4] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994.
- [5] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Intl. Conf. on Machine Learning*, Bari, Italy, 1996, pp. 148–156.
- [6] Y. Freund and R. E. Schapire, "Game theory, on-line prediction and boosting," in *Proc. 9th Annual Conf. on Computational Learning Theory*, Desenzano di Garda, Italy: ACM Press, 1996, pp. 325–332.
- [7] V. Boyarshinov and M. Magdon-Ismali, "Efficient Optimal Linear Boosting of a Pair of Classifiers," *IEEE Trans. Neural Networks*, vol. 18, no. 2, pp. 317–328, 2007.

- [8] R. E. Schapire and Y. Singer, “Improved boosting algorithms using confidence-rated predictions,” *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.
- [9] J. Friedman, T. Hastie and R. Tibshirani, “Additive Logistic Regression: a Statistical View of Boosting,” *Annals of Statistics*, vol. 28, pp. 337–374, 2000.
- [10] C. Cachin, “Pedagogical pattern selection strategies,” *Neural Networks*, vol. 7, no. 1, pp. 175–181, 1994.
- [11] P. E. Hart, “The condensed nearest neighbor rule,” *IEEE Trans. Information Theory*, vol. 14, no. 3, pp. 515–516, 1968.
- [12] P. W. Munro, “Repeat until bored: A pattern selection strategy,” in *Advances in Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson and R. Lippman, eds. San Mateo, CA: Morgan Kaufmann, 1992, pp. 1001–1008.
- [13] M. Plutowski and H. White, “Selecting concise training sets from clean data,” *IEEE Trans. Neural Networks*, vol. 4, no. 2, pp. 305–318, 1993.
- [14] M. Wann, T. Hediger, and N. N. Greenbaum, “The influence of training sets on generalization in feed-forward neural networks,” in *Proc. Intl. Joint Conf. on Neural Networks*, vol. III, San Diego, CA, 1990, pp. 137–142.
- [15] B. T. Zhang, “Accelerated learning by active example selection,” *Intl. Journal of Neural Systems*, vol. 5, no. 1, pp. 67–75, 1994.
- [16] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, “Boosting the margin: A new explanation for the effectiveness of voting methods,” *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.
- [17] J. Arenas-García, A. R. Figueiras-Vidal, and A. J. C. Sharkey, “The beneficial effects of using multi net systems that focus on hard patterns,” in *Multi Classifier Systems (Proc. 4th Intl. Workshop)*, T. Windeatt and F. Rolli, eds., Surrey, UK: Springer-Verlag (LNCS), 2003, pp. 45–54.
- [18] A. Lyhiyaoui, M. Martínez-Ramón, I. Mora-Jiménez, M. Vázquez-Castro, J. L. Sancho-Gómez, and A. R. Figueiras-Vidal, “Sample selection via clustering to construct support vector-like classifiers,” *IEEE Trans. Neural Networks*, vol. 10, no. 6, pp. 1474–1481, 1999.
- [19] R. García-Marcíel, I. Mora-Jiménez, and A. R. Figueiras-Vidal, “Improving kernel-based classifiers by guided dynamic sample selection,” in *Proc. 13th Intl. Conf. Artificial Neural Networks in Engineering*, A. Press, ed., St. Louis, MO, 2003, pp. 27–32.
- [20] J. Sklansky and L. Michelotti, “Locally trained piecewise linear classifiers,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 2, pp. 101–111, 1980.
- [21] L. Breiman, “Combining predictors,” in *Combining Artificial Neural Nets. Ensemble and Modular Multi-Net Systems*, A. J. C. Sharkey, ed., London, UK: Springer-Verlag, 1999, pp. 31–50.

- [22] C. J. C. Burges., “A tutorial on support vector machines for pattern recognition,” *Data Mining Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [23] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY: Springer-Verlag, 1995.
- [24] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, “An introduction to kernel-based learning algorithms,” *IEEE Trans. Neural Networks*, vol. 12, no. 2, pp. 181–201, 2001.
- [25] M. Lehtokangas, “Cascade-correlation learning for classification,” *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 795–798, 2000.
- [26] S. E. Falhman and C. Lebiere, “The cascade-correlation learning architecture,” in *Advances in Neural Information Processing Systems 2*, D. Touretzky, ed. San Mateo, CA: Morgan Kaufmann, 1990, pp. 524–532.
- [27] G. Rätsch, T. Onoda, and K. R. Müller, “Soft margins for Adaboost,” *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.
- [28] C. Rudin, I. Daubechies, and R. E. Schapire, “The dynamics of AdaBoost: Cyclic behavior and convergence of margins,” *Journal of Machine Learning Research*, vol. 5, pp. 1557–1595, 2004.
- [29] G. Rätsch and M. K. Warmuth., “Efficient margin maximizing with boosting,” *Journal of Machine Learning Research*, vol. 6, pp. 2131–2152, 2005.
- [30] C. Rudin, R. E. Schapire, and I. Daubechies, “Boosting based on a smooth margin,” in *Proc. 17th Annual Conf. on Computational Learning Theory*, Banff, Canada, 2004, pp. 502–517.
- [31] V. Gómez-Verdejo, M. Ortega-Moral, J. Arenas-García, and A. R. Figueiras-Vidal, “Boosting by weighting critical and erroneous samples,” *Neurocomputing*, vol. 69, pp. 679–685, 2006.
- [32] V. Gómez-Verdejo, M. Ortega-Moral, J. P. Cabrera, J. Arenas-García, and A. R. Figueiras-Vidal, “Boosting by emphasizing boundary samples,” in *Proc. of the Learning’04 Intl. Conf.*, Elche, Spain, 2004, pp. 67–72.
- [33] V. Gómez-Verdejo, M. Ortega-Moral, J. Arenas-García, and A. R. Figueiras-Vidal, “Boosting by weighting boundary and erroneous samples,” in *Proc. 13th European Symp. on Artificial Neural Networks*, Bruges, Belgium, 2005, pp. 85–90.
- [34] R. E. Schapire, “The strenght of weak learnability,” *30th Ann. Symp. Foundations on Computer Sci.*, 1989, pp. 28–33.
- [35] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos and S. Z. Li, “Ensemble-based discriminant learning with Boosting for face recognition,” *IEEE Trans. Neural Networks*, vol. 17, no. 1, pp. 166–178, 2006.
- [36] T. Windeatt, “Accuracy/Diversity and Ensemble MLP Classifier Design,” *IEEE Trans. Neural Networks*, vol. 17, no. 5, pp. 1194–1211, 2006.
- [37] D. Opitz and R. Maclin, “Popular ensemble methods: an empirical study,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

- [38] H. Schwenk and Y. Bengio, “Boosting neural networks,” *Neural Computation*, vol. 12, no. 8, pp. 1869–1887, 2000.
- [39] V. Gómez-Verdejo, J. Arenas-García, M. Ortega-Moral, and A. R. Figueiras-Vidal, “Designing RBF classifiers for weighted boosting,” in *Proc. of the Intl. Joint Conf. on Neural Networks 2005*, Montreal, Canada, 2005, pp. 1057–1062.
- [40] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Addison-Wesley, 1984.
- [41] R. Meir and G. Rätsch., “An introduction to boosting and leveraging,” in *Advanced Lectures on Machine Learning*, S. Mendelson and A. Smola, eds., New York, NY: Springer -Verlag, 2003, pp. 119–184.
- [42] V. Vapnik and A. Chervonenkis, “On the uniform convergence of relative frequencies of events to their probabilities,” *Theory of Probability and its Applications*, vol. 16, no. 2, pp. 264–280, 1971.
- [43] G. Rätsch, *Robust boosting via convex optimization: Theory and applications*, Ph.D. dissertation, University of Potsdam, Department of Computer Science, Potsdam, Germany, 2001.
- [44] J. von Neumann, “Zur Theorie der Gesellschaftsspiele,” *Mathematics Annals*, vol. 100, pp. 295–320, 1928.
- [45] J. T. Kwok, “Moderating the output of support vector classifiers,” *IEEE Trans. Neural Networks*, vol. 10, pp. 1018–1031, 1999.
- [46] B. D. Ripley, “Neural networks and related methods for classification (with discussion),” *Journal of the Royal Statistical Society. Series B*, vol. 56, pp. 409–456, 1994.
- [47] C. L. Blake and C. J. Merz, “UCI repository of machine learning databases,” University of California, Irvine, Dept. of Information and Computer Sciences: <http://www.ics.uci.edu/~mllearn/MLRepository.html>, 1998.
- [48] P. Alinat, “Periodic Progress Report 4, ROARS Project ESPRIT II - Number 5516,” in *Technical Thomson Report TS ASM 93/S/EGS/NC/079*, 1993.
- [49] A. Ruiz and P. E. L. de Teruel, “Nonlinear kernel-based statistical pattern analysis,” *IEEE Trans. Neural Networks*, vol. 12, no. 1, pp. 16–32, 2001.
- [50] F. Pérez-Cruz, “IRWLS Matlab toolbox to solve the SVM for pattern recognition and regression estimation,” <http://www.tsc.uc3m.es/~fernando/>, 2002.

APPENDIX I

To obtain an analytical expression to calculate output weights when the mixed emphasis function (11) is used, we keep the criterion used by the classical RA algorithm; concretely, at each round, we select the output weight value (denoted as β_t) that minimizes the training error bound given by (4). However, as

explained in [8], when $o_t(\mathbf{x}) \in [-1, 1]$, the output weight can only be calculated numerically. As in [8], to obtain an analytical expression for β_t , we will minimize the upper bound given in (5), but replacing α_t with the β_t coefficient used for the new scheme:

$$B_t \leq \frac{1}{L} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l)d_l] \left\{ \frac{1 + u_t(\mathbf{x}_l)}{2} \exp(-\beta_t) + \frac{1 - u_t(\mathbf{x}_l)}{2} \exp(\beta_t) \right\} \quad (\text{I-38})$$

where we use $u_t(\mathbf{x}_l) = o_t(\mathbf{x}_l)d_l$ to simplify the following formulae; the equality holds for $u_t(\mathbf{x}_l) = \pm 1$.

To obtain the minimum of this bound, let us take its first derivative with respect to β_t , and equal the result to 0:

$$\frac{1}{L} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l)d_l] \left\{ -\frac{1 + u_t(\mathbf{x}_l)}{2} \exp(-\beta_t) + \frac{1 - u_t(\mathbf{x}_l)}{2} \exp(\beta_t) \right\} = 0 \quad (\text{I-39})$$

from which

$$\begin{aligned} \exp(2\beta_t) &= \frac{\sum_{l=1}^L [1 + u_t(\mathbf{x}_l)] \exp[-f_{t-1}(\mathbf{x}_l)d_l]}{\sum_{l=1}^L [1 - u_t(\mathbf{x}_l)] \exp[-f_{t-1}(\mathbf{x}_l)d_l]} \\ &= \frac{LB_{t-1} + \sum_{l=1}^L u_t(\mathbf{x}_l) \exp[-f_{t-1}(\mathbf{x}_l)d_l]}{LB_{t-1} - \sum_{l=1}^L u_t(\mathbf{x}_l) \exp[-f_{t-1}(\mathbf{x}_l)d_l]} \end{aligned} \quad (\text{I-40})$$

Therefore, introducing

$$\delta_t = \frac{1}{LB_{t-1}} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l)d_l] o_t(\mathbf{x}_l)d_l \quad (\text{I-41})$$

and taking logarithms in (I-40), we obtain the desired expression for β_t :

$$\beta_t = \frac{1}{2} \ln \left(\frac{1 + \delta_t}{1 - \delta_t} \right) \quad (\text{I-42})$$

APPENDIX II

Let us rewrite (I-38) in a more convenient form:

$$\begin{aligned} B_t &\leq \frac{1}{L} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l)d_l] \left\{ \frac{1 + u_t(\mathbf{x}_l)}{2} \exp(-\beta_t) + \frac{1 - u_t(\mathbf{x}_l)}{2} \exp(\beta_t) \right\} \\ &= \frac{1}{2L} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l)d_l] \{ [\exp(-\beta_t) + \exp(\beta_t)] + u_t(\mathbf{x}_l) [\exp(-\beta_t) - \exp(\beta_t)] \} \end{aligned} \quad (\text{I-43})$$

Now, using (15), we have

$$\exp(-\beta_t) + \exp(\beta_t) = \sqrt{\frac{1-\delta_t}{1+\delta_t}} + \sqrt{\frac{1+\delta_t}{1-\delta_t}} = \frac{1-\delta_t+1+\delta_t}{\sqrt{1-\delta_t^2}} = \frac{2}{\sqrt{1-\delta_t^2}} \quad (\text{I-44})$$

$$\exp(-\beta_t) - \exp(\beta_t) = \sqrt{\frac{1-\delta_t}{1+\delta_t}} - \sqrt{\frac{1+\delta_t}{1-\delta_t}} = \frac{1-\delta_t-(1+\delta_t)}{\sqrt{1-\delta_t^2}} = \frac{-2\delta_t}{\sqrt{1-\delta_t^2}} \quad (\text{I-45})$$

Introducing (I-44) and (I-45) into (I-43), and applying some elementary manipulations, we arrive to

$$\begin{aligned} B_t &\leq \frac{1}{2L} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l)d_l] \left\{ \frac{2}{\sqrt{1-\delta_t^2}} + u_t(\mathbf{x}_l) \frac{-2\delta_t}{\sqrt{1-\delta_t^2}} \right\} \\ &= \frac{1}{\sqrt{1-\delta_t^2}} \frac{1}{L} \sum_{l=1}^L \exp[-f_{t-1}(\mathbf{x}_l)d_l] \{1 - u_t(\mathbf{x}_l)\delta_t\} \\ &= \frac{1}{\sqrt{1-\delta_t^2}} \{B_{t-1} - B_{t-1}\delta_t^2\} = \sqrt{1-\delta_t^2} B_{t-1} \end{aligned}$$

Finally, the recursive application of this result, together with the definition $\delta^2 = \min_{t'=1, \dots, T} \{\delta_{t'}^2\}$, leads to the desired expression for the upper bound on B_t :

$$B_t \leq \prod_{t'=1}^t \sqrt{1-\delta_{t'}^2} \leq (1-\delta^2)^{\frac{t}{2}} \quad (\text{I-46})$$

APPENDIX III

To obtain (32) and (33), consider first that, when the mixed emphasis function is used, (31) can be straightforwardly converted into (32) with

$$E_{\lambda,t}^\delta = \sum_{l=1}^L D_{\lambda,t}(\mathbf{x}_l) o_t(\mathbf{x}_l) d_l \quad (\text{I-47})$$

Now, note that the mixed emphasis function can be rewritten in the following manner:

$$\begin{aligned} D_{\lambda,t}(\mathbf{x}_l) &= \frac{1}{Z_{\lambda,t}} \exp \left\{ \lambda [f_{t-1}(\mathbf{x}_l) - d_l]^2 - (1-\lambda) f_{t-1}^2(\mathbf{x}_l) \right\} \\ &= \frac{1}{Z_{\lambda,t}} \exp[-f_{t-1}(\mathbf{x}_l)d_l] \exp \left\{ 2(\lambda - 0.5) \left[f_{t-1}(\mathbf{x}_l) - \frac{d_l}{2} \right]^2 \right\} \exp \left\{ (\lambda - 0.5) \frac{d_l^2}{2} \right\} \end{aligned} \quad (\text{I-48})$$

Finally, introducing the above expression into (I-47), and using the fact that the last exponential in (I-48) is a constant, we have

$$E_{\lambda,t}^\delta = \tilde{Z} \sum_{l=1}^L \exp \left\{ 2(\lambda - 0.5) \left[f_{t-1}(\mathbf{x}_l) - \frac{d_l}{2} \right]^2 \right\} \exp[-f_{t-1}(\mathbf{x}_l)d_l] o_t(\mathbf{x}_l) d_l \quad (\text{I-49})$$

\tilde{Z} being a constant, from which, apart from constant terms, (33) is readily obtained using the definition in (34).

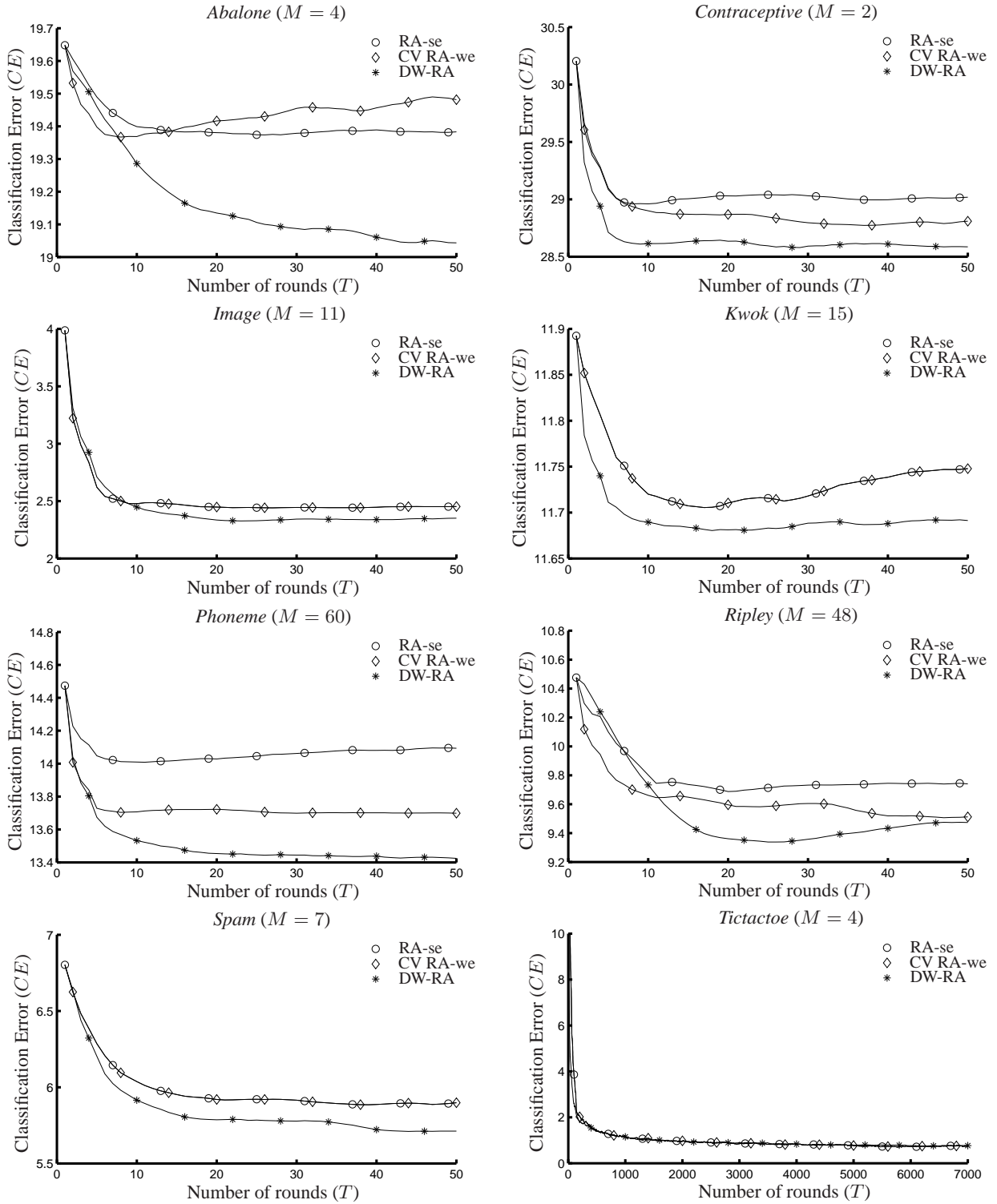


Fig. 1. CE convergence of RA-se, CV RA-we and DW-RA with fixed base learner complexity. The value of M used for each problem is indicated above each subfigure.

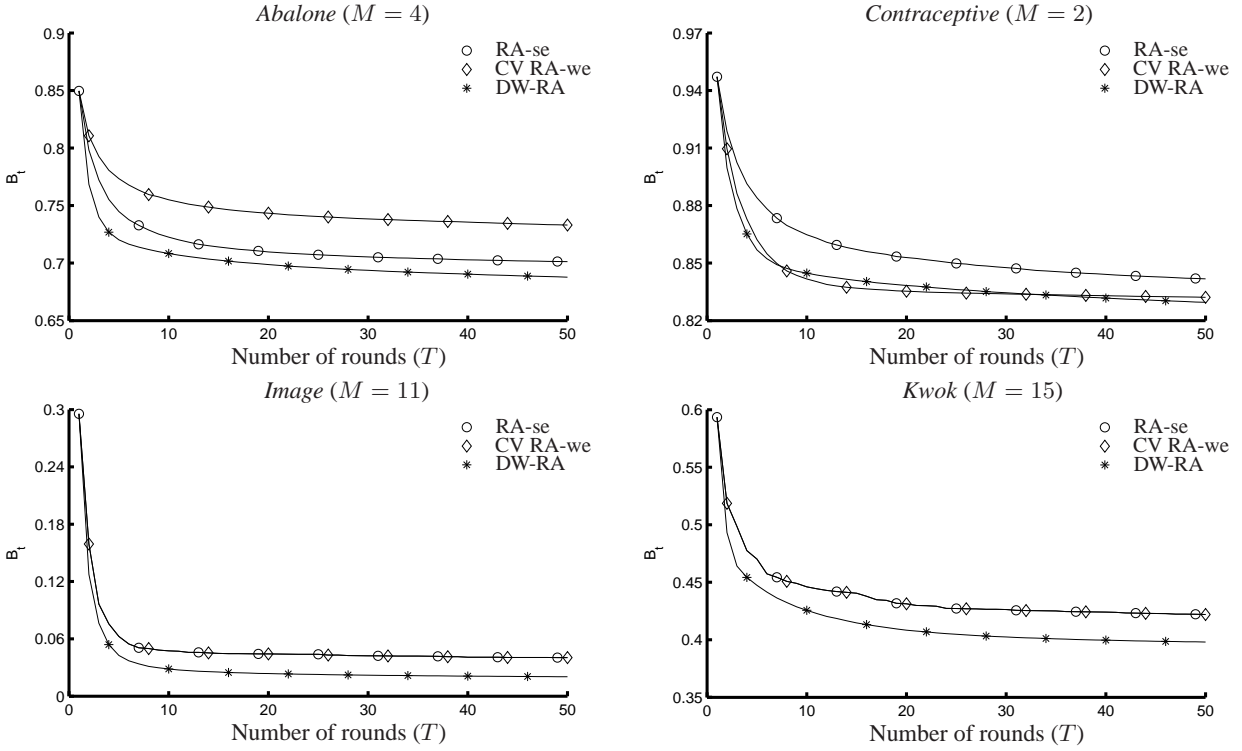


Fig. 2. Evolution of bound B_t as a function of the number of learners. RA-se, CV RA-we and DW-RA are constructed with fixed base learner complexity.

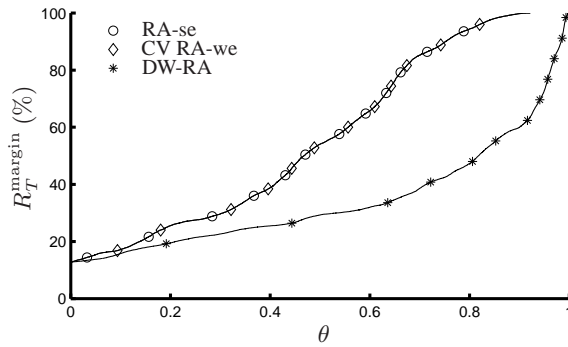


Fig. 3. Margin distribution for Ripley ($M = 48$).

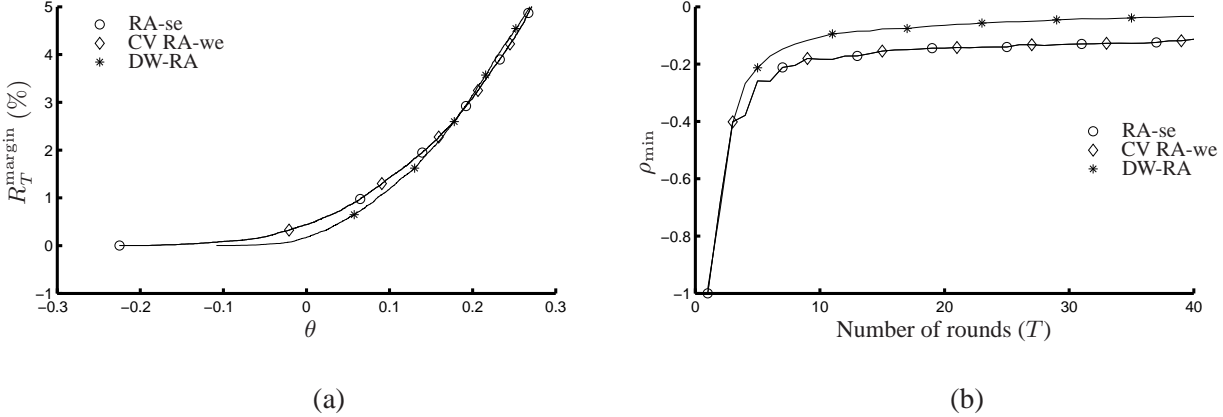


Fig. 4. ρ_{\min} values presented by the different approaches in *Image* for $M = 11$; (a) Margin distribution around zero; (b) Evolution of ρ_{\min} with the number of rounds

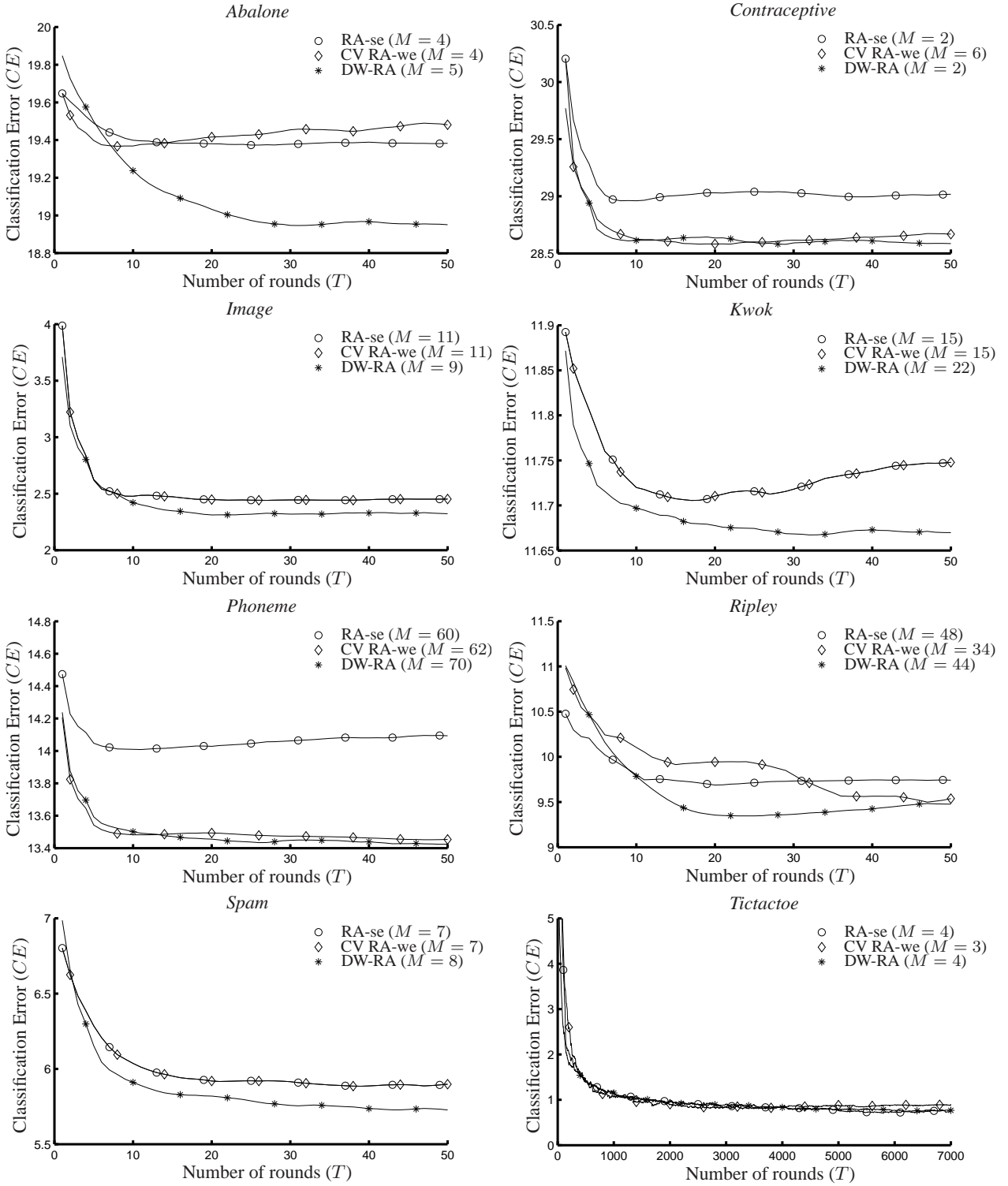


Fig. 5. CE convergence of RA-se, CV RA-we and DW-RA for different data sets.