Real AdaBoost with Gate Controlled Fusion

Efraín Mayhua-López, Vanessa Gómez-Verdejo, and
Aníbal R. Figueiras-Vidal, *Senior Member, IEEE*

***Abstract*—Real AdaBoost (RAB) ensembles have demonstrated exceptional classification capabilities, plausibly because they construct and combine weak learners that paying attention to samples that are more difficult to label at each step. However, they use a fixed linear combination of these learners, which can be a limitation for their expressive power. On the other hand, Mixtures of Experts (MoE) can easily include powerful gates to combine very simple learners, although they suffer from learning difficulties for classification purposes.**

In this paper, we propose to increase the capabilities of standard RAB architectures replacing their linear combinations by a fusion controlled by a gate with fixed kernels. Experimental results in a series of well-known benchmark problems support the effectiveness of this approach to improve classification performance. Although the need of cross-validation processes obviously leads to higher training computational effort, operation load is never much higher, and in some cases it results even lower than that of competitive RAB schemes.

***Index Terms*—Classification, neural networks, ensembles, Real AdaBoost, Mixtures of Experts.**

I. Introduction

A. Preliminaries

Ensemble classifiers are deserving much interest because they allow a relatively easy design and offer a high performance. Some of their architectures permit to understand how decision are made better than standard neural networks. Detailed discussions of their general characteristics and those of their main families can be found in [1]–[3].

Among ensemble classifiers, boosting algorithms exhibit exceptional capabilities. Based on the potential of combining weak learners, they were formulated for the first time under a filtering form [4], after which the principled constructive technique with binary output learners called AdaBoost (AB) was proposed [5]. Real AdaBoost (RAB) [6] extended this idea to real-valued output learners. From its very beginning, boosting was considered a fundamental contribution; Breiman said that it was the most significant development of the 1990s decade in designing classifiers.

A singular characteristic of boosting techniques is their resistance to overfitting, experimentally found in several works, such as [7]–[12]. A number of authors, including the proposers of boosting [6] [13], considers that this advantage comes from their connection with Margin Maximization (MM). On the contrary, Breiman [14] [15] supports that it is due to using resampling and weak learners. In any case, other experimental studies discovered overfitting —see [16]–[19], for example—, mainly in cases in which there are samples that result clearly missclassified. This fact prompted the appearance of a series of works introducing methods to reduce this problem: Freund himself [20] proposed to eliminate the samples of this type; in [15], [21]–[23] different forms of soft-margin and regularization are analyzed. The consistency (convergence to Bayes risk) of regularized boosting is discussed in [24]; recently, the authors of [25] proved that AB is almost surely consistent if its growing is stopped early enough. Other approaches reducing the sensitivity of boosting algorithms with respect to clearly misplaced samples can be found in [26], where the data skewness is penalized to prevent the effects of these samples; in [27], which maximizes the mean and minimizes the variance of the margin; in [28] [29], that replace the MM cost terms by adjustable combinations of quadratic error and proximity to the decision border; and in [30], which uses controlled subsampling to reduce the problem.

There is a large quantity of boosting algorithms that are modifications of basic AB and RAB. To mention just a few: Those based on the idea of leveraging [31], presented in [32] [33]; DOOM (Direct Optimization Of Margin) [34]; AnyBoost and MarginBoost [35]; formulations to accommodate linear classifiers as learners [36] [37]; methods that exploit the vision from a logistic regression perspective [38] and using stochastic search for residual fitting [39]; cost-sensitive algorithms [40]; procedures for optimally boosting classifiers [41]; and algorithms resulting from dual formulations [42]. There are versions for imbalanced data problems [43] and for semisupervised learning [44] [45]. Boosting has been used to select kernels [46] [47]. Finally, it is worth saying that there are recent works that combine boosting with other techniques —with nonlinear projections [48] or with Rotation Forests [49], for example—.

Another important family of ensembles that has deserved attention is Mixtures of Experts (MoE) [50]. When considering regression problems, they are based on assuming a Gaussian mixture model with a linear mean for the "a posteriori" probability of the unknown variable and a softmax linear form for the mixing coefficients. Using Maximum Likelihood (ML) as the objective to parameterize these models seems to be enough to control overfitting. The formulation for classification is obtained by using the exponential form of the target binary distribution and logistic activations for the experts. Direct or Iterative Reweighted Least Squares versions of the Expectation-Maximization algorithm can be used to train the corresponding schemes [51] [52]. More expressive capacity is available from hierarchical architectures [51]–[54], using Multilayer Perceptrons (MLPs) as learners [55], introducing

The authors are with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés (Madrid), Spain (email: {emayhua,vanessa,arfv}@tsc.uc3m.es).

This work has been supported by Spanish MEC grant TEC2008-02473/TEC.

more powerful gates such as MLPs [56]–[58] or including attentional mechanisms for fusion [59].

MoE are related to stacking [60], and there are several modifications of them, such as those presented in [61] [62]. Among these modifications, there are constructive versions [54] [63] as well as temporal schemes [64], and procedures that use Dirichlet processes mixtures or as priors [65] [66]. Parameterization algorithms using global searches are proposed in [67]–[69]. The consistency of MoE classifiers using logistic regression units has also been studied in [70].

MoE offered satisfactory results in may practical applications such as speech recognition [53] [62], time series analysis [56]–[58] [64] [71] [72], handwritten character recognition [61] [62] [68], AIDS prognosis [73], and face recognition [74]. Nevertheless, contributions to MoE literature are less frequent in recent years, maybe because the great success of other ensemble families, such as boosting methods. However, MoE bases are solid, and MoE concepts, useful. They have been used to mix MM trained linear classifiers [75]; in [76], the authors reorder the MoE basic formula to obtain a similar but more compact architecture which can be trained with MM algorithms.

B. The conceptual bases of the proposed ensemble architecture

The fusion scheme of AB's and RAB's weak learners is a linear combination by means of weights that minimize an exponential margin error and are easily calculated. This may limit the ensemble performance when using global weak learners -such as trees and simple MLPs, that are frequently boosted-, because the overall ensemble architecture is also purely global.

Using local learners can alleviate this limitation. Several studies have considered stumps as learners. In [77], learners specialize in different regions. Rätsch et al. [78] analyzed the effects of using local learners. In [79], localization is based on applying local likelihoods, while in [80] kernels are included in learners. A local error measure is applied in [81]. These approaches demonstrate some advantages, but it seems clear that using local learners does not serve "per se" to limit the negative effects of strongly emphasizing clearly erroneous samples.

On the contrary, MoE include a trainable gate, a more sophisticate and powerful fusion technique. Consequently, it is tempting to try to obtain a further increase of the capabilities of boosting schemes by replacing their linear combination by a trainable gate. In fact, this was explored in [82] as an iterative construction of a functional convex combination, but under an ML parameterization.

Following this idea, in this paper a new boosting approach including a gate network in the learner fusion process is introduced. This new method, called Gate Controlled Fusion RAB (GCF-RAB), will retain the good properties of standard RAB algorithm but it would provide the ensemble improved convergence and generalization capabilities. As we will explain later, a local gate with intermediate expressive power is an attractive option, because it can serve to reduce the effects of overweighting clearly wrongly classified samples. Note that the gate has to be trained just for fusion purposes; in other case, we would tend to work with stronger learners. Since the excellent results provided by arcing ensembles (including boosting schemes) can be attributed to emphasizing weak learners training and, after it, combining them, to explore how a locally gated RAB machine works is definitely interesting. In fact, we obtained promising preliminary results following this research line [83].

The rest of the paper is organized as follows: In Section II, we introduce the general structure of the proposed classification ensemble, discussing how to train its learnable parameters and different possibilities to select its design parameters by means of cross-validation (CV). Section III describes the experimental framework, shows the simulation results, and comparatively discusses the performance of the new ensemble, without forgetting sensitivity and computational load aspects. Finally, Section IV presents the conclusions of our work and suggests some further research directions.

II. Gate Controlled Fusion Real AdaBoost (GCF-RAB)

A. The basic structure

To avoid serious algorithmic difficulties in training and an excessive operation load, a common gate body consisting of Gaussian kernels with selected centroids will be used for all the fusion steps, its output weights being trained for each epoch.

We will consider binary problems. Then, we have the output of the ensemble of T learners is

$$F_T(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t(\mathbf{x}) f_t(\mathbf{x}) \tag{1}$$

where α_t and $f_t \in [-1, 1]$ are the overall gate and the base learners output at round t, respectively. Decision is made according to

$$\hat{d}(\mathbf{x}) = \text{sgn}\left[F_T(\mathbf{x})\right] \tag{2}$$

where sgn is the standard sign function.

Base learners are trained in the conventional RAB form using the labeled data set $\left\{\mathbf{x}^{(l)}, d^{(l)}\right\}_{l=1}^{L}$, where $\left\{\mathbf{x}^{(l)}\right\}$ are the samples and $\left\{d^{(l)} \in \{\pm 1\}\right\}$ their targets, respectively. Each base learner is parameterized to minimize the weighted squared error

$$E_t = \sum_{l=1}^{L} D_t(l) \left[d^{(l)} - f_t(\mathbf{x}^{(l)})\right]^2 \tag{3}$$

D_t being the well-known RAB emphasis function

$$D_{t+1}(l) = \frac{D_t(l) \text{exp}\left[-\alpha_t(\mathbf{x}^{(l)}) f_t(\mathbf{x}^{(l)}) d^{(l)}\right]}{Z_t} \tag{4}$$

where Z_t is a normalization factor to force $\sum_{l=1}^{L} D_{t+1}(l) = 1$, and the process starts with $D_1(l) = 1/L, \ \forall l$.

As above said,

$$\alpha_t(\mathbf{x}) = \mathbf{w}_t^T \mathbf{a}(\mathbf{x}) \tag{5}$$

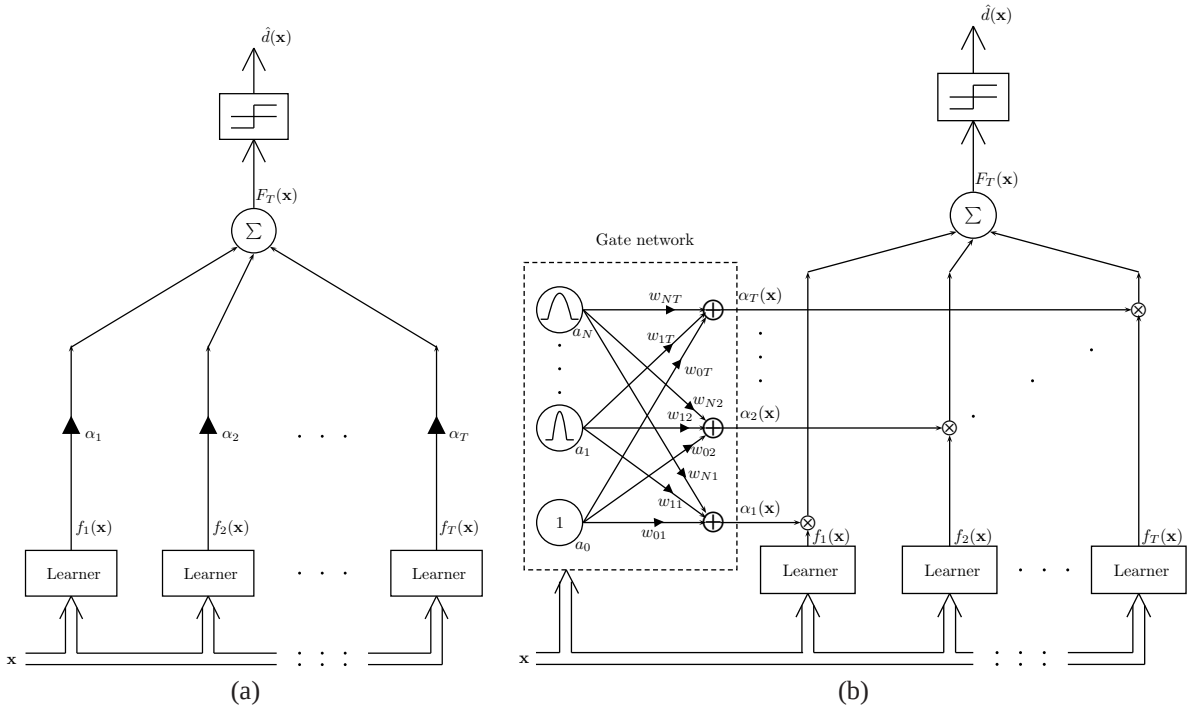

Fig. 1. (a) Real AdaBoost architecture; (b) Gate Controlled Fusion Real AdaBoost architecture.

where $\mathbf{w}_t$ is the output weight vector of the gate for base learner f_t, and $\mathbf{a}(\mathbf{x}) = [a_0(\mathbf{x}),\ a_1(\mathbf{x}),\ \ldots\ , a_N(\mathbf{x})]^T$ the output vector of the Gaussian functions of the gate body, having elements

$$a_n(\mathbf{x}) = \begin{cases} 1, & n = 0 \\ \exp\left(-\|\mathbf{x} - \mathbf{c}_n\|^2 / 2{\sigma_n}^2\right), & 1 \le n \le N \end{cases} \tag{6}$$

$\{\mathbf{c}_n\}$ being the centroids, that are separately selected, and $\left\{{\sigma_n}^2\right\}$ the dispersion parameters, that can be selected as explained later. Fig.1 shows the proposed GCF-RAB architecture, comparing it with the standard RAB. Learning $\mathbf{w}_t$ is carried out to minimize the overall cost function of the RAB algorithm

$$B_t = \frac{1}{L}\sum_{l=1}^{L} \exp\left[-d^{(l)} F_t(\mathbf{x}^{(l)})\right] \tag{7}$$

where $F_t(\mathbf{x})$ is the partial ensemble obtained at epoch t

$$F_t(\mathbf{x}) = \sum_{t'=1}^{t} \alpha_{t'}(\mathbf{x}) f_{t'}(\mathbf{x}) = F_{t-1}(\mathbf{x}) + \alpha_t(\mathbf{x}) f_t(\mathbf{x}) \tag{8}$$

The reason for adopting this cost is to retain the good generalization properties of the classical RAB algorithm.

Since

$$\frac{\partial \exp\left[-d^{(l)} F_t(\mathbf{x}^{(l)})\right]}{\partial \mathbf{w}_t} =$$

$$= -d^{(l)} f_t(\mathbf{x}^{(l)}) \exp\left[-d^{(l)} F_t(\mathbf{x}^{(l)})\right] \mathbf{a}(\mathbf{x}^{(l)}) \tag{9}$$

the stochastic descent algorithm

$$\mathbf{w}_t(l'+1) = \mathbf{w}_t(l') + \eta d^{(l')} f_t(\mathbf{x}^{(l')}).$$

$$.\exp\left[-d^{(l')} F_t(\mathbf{x}^{(l')})\right] \mathbf{a}(\mathbf{x}^{(l')}) \tag{10}$$

can be applied to samples cyclically ordered (index l').

B. Selecting centroids

Training centroids and dispersion of exponentials has been considered a difficult task, and most the corresponding machine designs have been based on defining them by means of a separate procedure. For regression, after the pioneering proposal of Moody and Darken [84], suggesting to select K-means vectors as centroids, a large number of proposals appeared: A sequential selection based on Orthogonal Least Squares (OLS) [85], constructive methods like resource allocation [86] and growing cell structure [87], and some refined clustering algorithms [88] [89], among others.

In the case of decision machines, things are different because border location is important. In fact, although RAB emphasizes erroneous samples, the resulting machines focused mainly in examples that are near the classification borders [90]. [91] and [92] select centroids among clustering representative vectors; considering methods that select sample vectors as exponential function centroids, [93] presents a series of possibilities after a first cluster selection, while [94] proposes to use just the support vectors of Support Vector Machine designs. The idea of using K-Nearest Neighbors (K-NN) algorithms to select samples according to their proximity to the border and their easy classification appears in [95]–[98]; here, we will start from it for designing our machine ensembles.

In [95], Shin and Cho introduce two quantitative measures to be used for selecting data, "proximity", $pr(\mathbf{x}^{(l)})$, and

"correctness", $co(\mathbf{x}^{(l)})$. These measures are obtained from K-NN based estimates of the probabilities of $\mathbf{x}^{(l)}$ belonging to class j ($j \in \{\pm 1\}$)

$$P_j(\mathbf{x}^{(l)}) = \frac{K_j}{K}$$

K and K_j being the total number of nearest neighbors considered and the number of these belonging to class j, respectively.

It is obvious that

$$pr(\mathbf{x}^{(l)}) = \sum_j P_j(\mathbf{x}^{(l)}) \log_2 \frac{1}{P_j(\mathbf{x}^{(l)})} \tag{11}$$

is higher for samples whose neighbors have mixed labels, i.e., for samples that are near the classification border, while

$$co(\mathbf{x}^{(l)}) = P_{d^{(l)}}(\mathbf{x}^{(l)}) \tag{12}$$

is the estimated probability of a correct classification of $\mathbf{x}^{(l)}$. Preselecting those samples that offer

$$pr(\mathbf{x}^{(l)}) > 0 \quad \text{and} \quad co(\mathbf{x}^{(l)}) \geq \frac{1}{2} \tag{13}$$

is a satisfactory mode of selecting reasonable (near to the border and relatively easy to classify) candidates to serve as Gaussian centroids.

To complete the selection of centroids avoiding to include samples that appear in the same regions of the observation space, we apply the Adaptive Pattern Classifier III clustering algorithm [99] ideas. For each class, the first centroid is the preselected sample which shows the highest proximity value; all the samples in a sphere of radius h around it are eliminated, and the process is iterated until no more samples are available.

The value of h can be obtained by means of CV of R in the expression

$$h = R \frac{1}{L} \sum_{l=1}^{L} \left\| \mathbf{x}^{(l)} - \mathbf{x}_{\mathrm{NN}}^{(l)} \right\|_2 \tag{14}$$

where $\mathbf{x}_{\mathrm{NN}}^{(l)}$ indicates the nearest neighbor of $\mathbf{x}^{(l)}$.

With respect to the value of K, a possible option is to select it by CV. To reduce the design computational effort, the empirical procedure of [96] can be used. This procedure consists of applying an 1-NN classifier on the training set; then, K is iteratively increased until the number of preselected samples according to (13) is not less than $2L\hat{P}_e$, where $\hat{P}_e$ is the 1-NN estimate of the classification error probability.

C. Selecting the dispersion parameters

We will consider two different strategies to select the dispersion parameters:

- The first is accepting dispersion values that are proportional to the average Euclidean distance between $\mathbf{c}_n$ and all samples that are nearer to it than to the rest of centroids, i.e.

$$\sigma_n = r \frac{1}{card\ C_n} \sum_{\mathbf{x}^{(l)} \in C_n} \left\| \mathbf{x}^{(l)} - \mathbf{c}_n \right\|_2 \tag{15}$$

  where C_n is the corresponding set of samples and $card$ indicates cardinality. Scale factor r can be determined by means of CV.
- The second directly uses as dispersion the sphere radius (h) of the APC-III algorithm (which also requires a CV process).

III. Experiments and their discussion

In order to evaluate the performance of the proposed method, we will apply it to some well-known benchmark problems with different characteristics (size, dimension, and difficulty) and compare the obtained results with those provided by a standard RAB ensemble.

A. Benchmark problems

We consider eight problems. Two of them are synthetic, Kwok [100] and Ripley [101], and the other six are real data sets, five from the UCI repository [102] (Abalone, Breast, Contraceptive, Hepatitis, and Ionosphere), and one from [103], Crabs. Table I shows their main characteristics (D: dimension; L_1/L_{-1}: number of samples) for the training and the test sets. Problems are named with their first two letters.

TABLE I
Main characteristics of the benchmark problems

Problem	D	L_1/L_{-1} Train	L_1/L_{-1} Test
Ab	8	1238/1269	843/827
Br	9	145/275	96/183
Co	9	506/377	338/252
Cr	7	59/61	41/39
He	19	70/23	53/9
Io	34	101/100	124/26
Kw	2	300/200	6120/4080
Ri	2	125/125	500/500

B. Versions of the GCF-RAB algorithm

We consider the two strategies presented above to define dispersion parameters, calling them S1 and S2. When K is selected according to the empirical rule given in [96], we indicate the corresponding reduced versions as S1(R) and S2(R), respectively.

C. Trainable parameters

Both RAB and GCF-RAB ensembles have Multi-Layer Perceptrons (MLPs) with one simple hidden layer of M neurons as base learners, M being established by means of CV. Each MLP is trained with the standard Back-Propagation algorithm to minimize (3), initializing all the weights at random values from a $[-0.2, 0.2]$ uniform distribution, while the learning rate for both layers linearly decrease from 0.01 to 0 along 100 epochs, that are enough to reach convergence. An 80/20 early stopping mode is applied to stop training.

With respect to gate weights, they are randomly initialized with values of a $[-0.05, 0.05]$ uniform distribution, and a

TABLE II
CLASSIFICATION ERROR RATES (CE) PROVIDED BY RAB ENSEMBLE AND THE DIFFERENT APPROACHES OF THE PROPOSED GCF-RAB ALGORITHM. THE NUMBER OF LEARNERS (T) AND THE PARAMETERS CHARACTERIZING EACH ENSEMBLE ARE ALSO INCLUDED.

		RAB	GCF-RAB				
			S1	S2	S1(R)	S2(R)	Omniscient
		M	$M/N/K/R/r$	$M/N/K/R$	$M/N/K/R/r$	$M/N/K/R$	$M/N/K/R/r$
Ab	$CE(\%)$	19.4 ± 0.02	**19.1** ± 0.3	19.3 ± 0.3	**19.1** ± 0.3	19.3 ± 0.3	19.0 ± 0.4
	T	31.2 ± 0.4	16.3 ± 0.4	30.5 ± 0.9	19.0 ± 0.4	29.2 ± 0.8	16.0 ± 0.5
	Param.	4	2/50/6/6.2/1	2/7/6/34.5	2/7/4/34.5/1	2/7/4/34.5	2/405/10/1.0/2.5
Br	$CE(\%)$	2.6 ± 0.4	**2.2** ± 0.3	2.3 ± 0.4	2.4 ± 0.3	2.4 ± 0.5	2.2 ± 0.3
	T	21.3 ± 4.2	16.1 ± 0.8	18.0 ± 1.2	18.2 ± 0.7	32.6 ± 1.0	16.4 ± 1.3
	Param.	6	2/18/8/11.3/3.5	2/18/2/1.0	2/4/3/24.2/1	6/7/3/11.3	2/10/4/11.3/3.5
Co	$CE(\%)$	29.0 ± 0.2	**27.4** ± 0.8	28.1 ± 0.8	28.2 ± 0.8	28.2 ± 0.8	27.4 ± 0.5
	T	33.7 ± 0.7	21.7 ± 1.1	29.7 ± 0.7	29.9 ± 1.4	30.4 ± 1.2	21.4 ± 0.5
	Param.	2	2/165/3.6/3.5	2/8/4/34.5	2/48/2/11.3/1	2/17/2/24.2	2/21/4/21.6/3.0
Cr	$CE(\%)$	**2.5** ± 0.0	**2.5** ± 0.0	**2.5** ± 0.0	**2.5** ± 0.0	**2.5** ± 0.0	2.5 ± 0.0
	T	11.1 ± 0.8	16.1 ± 0.5	8.1 ± 0.2	19.3 ± 0.6	31.5 ± 0.9	16.1 ± 0.4
	Param.	2	2/34/8/1.0/1.5	2/11/2/1.0	2/7/2/1.0/1	2/7/2/1.0	2/11/2/1.0/1
He	$CE(\%)$	8.9 ± 1.8	**8.1** ± 1.9	8.4 ± 1.8	8.5 ± 1.5	8.5 ± 1.7	7.7 ± 1.3
	T	22.2 ± 3.9	15.6 ± 1.8	18.7 ± 2.5	16.9 ± 1.9	27.5 ± 2.0	16.1 ± 1.1
	Param.	17	4/20/4/3.6/4	4/3/2/21.6	6/2/2/39.7/0.5	4/2/2/31.9	4/2/6/26.8/2
Io	$CE(\%)$	4.5 ± 0.9	**4.0** ± 1.0	4.2 ± 1.7	4.1 ± 1.5	**4.0** ± 1.6	3.8 ± 1.8
	T	22.2 ± 2.4	17.3 ± 1.7	24.9 ± 1.9	19.1 ± 5.4	26.9 ± 3.6	17.7 ± 4.3
	Param.	5	2/10/6/34.5/3	2/66/10/1.0	2/8/2/26.8/2.5	6/21/2/6.2	4/66/10/1.0/2.5
Kw	$CE(\%)$	**11.7** ± 0.01	**11.7** ± 0.2	**11.7** ± 0.1	**11.7** ± 0.2	**11.7** ± 0.2	11.7 ± 0.1
	T	29.3 ± 0.1	19.8 ± 1.1	29.0 ± 1.2	28.8 ± 1.2	36.3 ± 1.4	20.5 ± 1.1
	Param.	15	2/4/6/31.9/2	2/7/10/16.5	4/4/2/31.9/1	4/4/2/42.3	2/10/4/11.3/3.5
Ri	$CE(\%)$	9.7 ± 0.01	**8.5** ± 0.2	8.6 ± 0.3	9.1 ± 0.2	9.2 ± 0.3	8.4 ± 0.3
	T	28.9 ± 0.2	17.9 ± 0.6	25.6 ± 0.5	24.2 ± 2.3	33.4 ± 0.7	17.8 ± 0.5
	Param.	48	2/31/10/1.0/2	6/10/4/6.2	6/5/2/21.6/0.5	2/6/2/16.5	6/31/10/1.0/2

learning rate of 0.01 is used for their stochastic gradient updating 100 epochs are enough for convergence.

To stop the ensemble construction, we apply to RAB the criterion successfully used in [29], selecting T as the first value for which

$$\frac{\sum_{t=T-9}^{T} \alpha_t}{\sum_{t=1}^{T} \alpha_t} < C \tag{16}$$

where C is empirically set to 0.1. Similarly, we use for the GCF-RAB

$$\frac{\sum_{t=T-9}^{T} \gamma_t}{\sum_{t=1}^{T} \gamma_t} < C' \tag{17}$$

with C'= 0.3, where

$$\gamma_t = \sum_{l=1}^{L} D_t(l) F_t(\mathbf{x}^{(l)}) d^{(l)} \tag{18}$$

which, when the learners are not able of classifying the most emphasized (erroneous) data, takes low values and hardly changes.

D. Selection of design parameters

A 50 runs, 5-fold cross-validation (CV), is used to select the nontrainable parameters; these parameters are explored in the following margins:

- M (number of hidden units of MLPs): from 2 to 10 in unitary steps;
- K (number of nearest neighbors in non-reduced strategies): 2,4,6,8,10 and 12;
- R (scale factor for h): from 1 to 50 with 20 equal steps;
- r (scale factor for the first strategy to select dispersion parameters): from 0.5 to 5 in 0.5 steps.

No extensions of these margins are needed except for parameter M in the RAB algorithm and Ri problem, which needs an extension up to 60.

E. Results and their discussion

Table II shows the Classification Error (CE) rates obtained from averaging the results corresponding to 50 design runs (of MLPs) for each ensembles using the parameters given by the CV processes, that are also given, as well an the average number of learners (T). For CE and T, both average values and standard deviations are included. Best results are indicated in boldface. It can be seen in Table II that none of the proposed GCF-RAB algorithms offers a worse performance than the conventional RAB. As expectable, Version S1 gives the best results, with the highest differences when compared to RAB for problems Br, He, Io, and mainly Ri, while RAB only provides an equivalent performance for Cr and Kw. Version S2 offers slightly worse results than Version S1, but, we repeat, never worse than RAB. This quality reduction can be attributed to the fact of restricting Gaussian dispersion parameters to just the same value of h. Finally, (reduced) Versions S1(R) and S2(R) provide, in general, worse performance results than Versions S1 and S2, respectively (but S2(R) is better than S2 for Io), but, again, never worse than RAB, although the direct way of establishing K limits their quality.

So, we can conclude that, generally speaking, the idea of introducing a gate to combine learners' outputs under a RAB

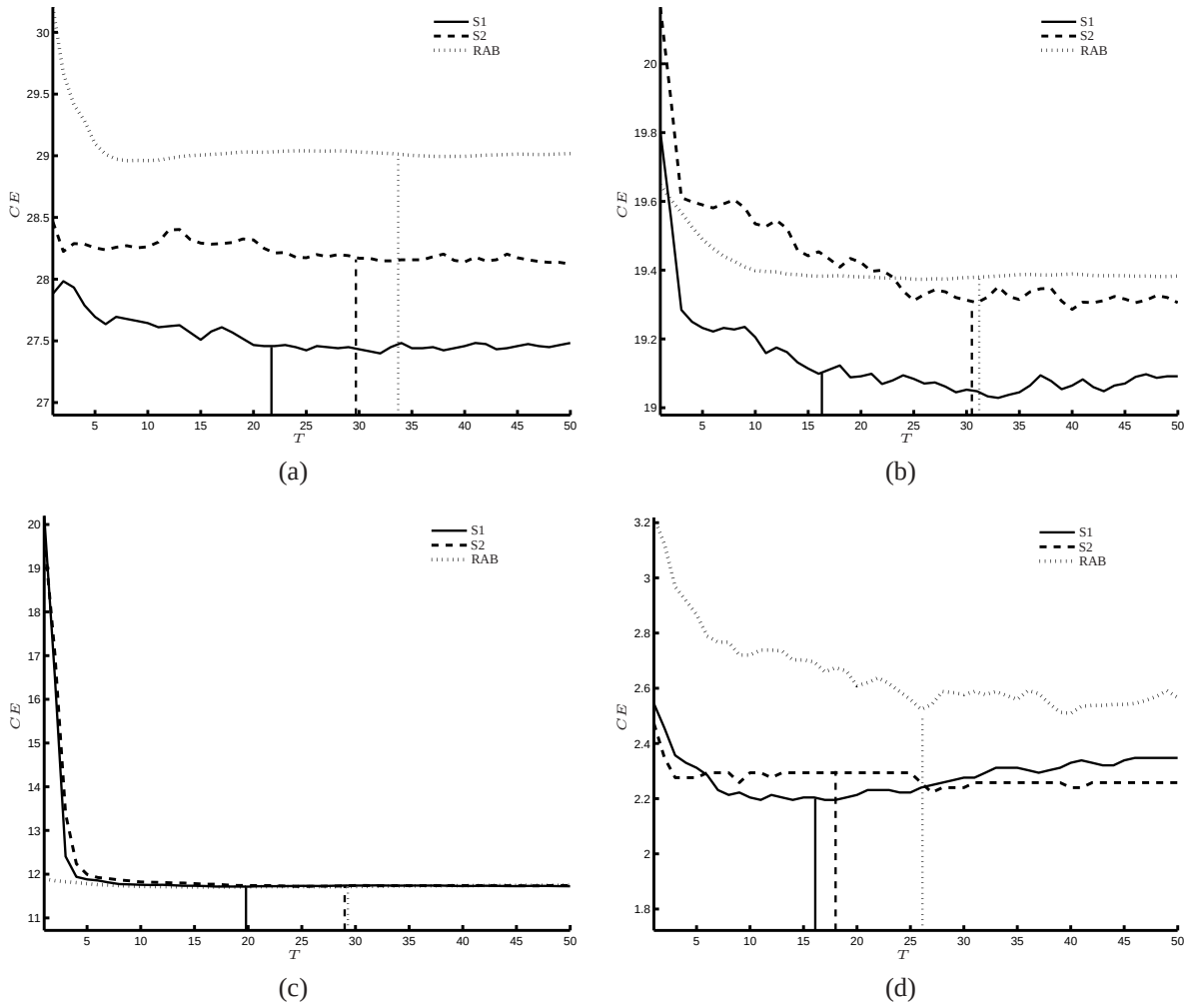

Fig. 2. Classification error rate (CE) evolution along training epochs for conventional RAB and GCF-RAB methods S1 and S2 for problems Co(a), Ab(b), Kw(c), and Br(d).

formulation serves to improve the performance of the resulting designs. Obviously, this is not for free, since training gate's weights and, specially, the costly CV process greatly increase the design computational effort (several orders of magnitude). This is the prize for improving the excellent results of RAB ensembles. Nevertheless, it can be seen also in Table II that, in many cases, the GCF-RAB designs require a lower (average) number of learners and a lower number of hidden neurons for the MLP learners. Since each complete learning element of a GCF-RAB consists not only on the corresponding MLP, but it includes the gate multiplications, more detailed calculations are needed to check if there is also computational advantage in the operation phase, i.e., when the ensembles are used to classify new patterns. These calculations are presented below.

F. A look to sensitivity problems

Although the proposed designs exhibit an excellent performance, it is adequate to analyze if the always delicate CV processes introduce some degree of lack of robustness with respect to the corresponding parameters. To study the sensitivity of the proposed designs with respect to all these parameters (M, h or N, K for S1 and S2, and r for S1 and S1(R)) is a difficult task. However, the possibility of using the concept of "omniscient" machines serves to carry out a sufficient analysis.

An "omniscient" machine —obviously not a valid design— is a machine where non-trainable parameters are selected according to the measured performance in the test set. If there are small differences between the corresponding parameter values and those values obtained from CV, the CV process has been fully successful. If there are differences but the performance is similar, it is a symptom of CV difficulties due to a relatively flat error surface considered as a function of the CV parameters, but the resulting designs can be considered acceptable —their performance is good enough—.

The last column of Table II shows the characteristics of the omniscient ensembles for the more delicate family of designs, Version S1. It is clear that a completely successful CV cannot be claimed, since there appear significant differences between parameter values. However, it is also evident that

performance differences are irrelevant for Br, Co, Cr, and Kw, nearly irrelevant for Ab and Ri, and not so important for He and Io. Consequently, we can conclude that the design methods that have been followed for the proposed GCF-RAB ensemble architectures show a satisfactory degree of robustness.

G. Convergence

Since we include a trainable fusion scheme, there is the possibility of having more intensive convergence problems with CGF-RAB methods than when applying conventional RAB. So, to check how convergence believes along training epochs is interesting.

In Fig. 2, we show how CE evolves for four representative problems:

- Co (He is qualitatively equivalent)
- Ab
- Kw (Cr and Io are qualitatively equivalent)
- Br (Ri is qualitatively equivalent)

for conventional RAB and proposed methods S1 and S2, that are the most relevant.

Generally speaking, convergence problems do not appear, and the (average) stopping epochs seem to be reasonable.

In the case of Co, convergence curves show the usual forms, and it is clear that the advantage of S1 and S2 appears very early. The stopping points reveal the advantage of the new methods with respect to conventional RAB with respect to the number of learners. However, let us remark that this does not mean that S1 and S2 learning is easier than training a conventional RAB. On the contrary, the needs of establishing the number of gate RBFs by CV and of training gate output weights at each step lead to a clearly higher computational effort to train S1 and S2 architectures. Even the operation load —i.e., the computational effort which is needed to classify a new sample— does not depend only on the number of learners, because there are calculations related to the gate. Since it can be observed in Table II that the complexity of learners is also usually lower for GCF-RAB methods than for conventional RAB, a more detailed evaluation is necessary. We present this evaluation later.

Considering problem Ab, the only remarkable difference is that the advantage of S1 with respect to RAB needs a relative high numbers of training epochs to appear.

Kw shows that initial RAB convergence is faster than those of S1 and S2. Although the stopping criterion leads to more epochs for conventional RAB, it can be seen that near-optimum designs could be obtained very early.

Finally, Br curves show that the optimal designs performances are as usual (S1 is better than S2, and S2 better than conventional RAB). However, is must be emphasized that S1 training shows a relatively clear effect of overfitting. This is probably due to the high expressive power of the learner-gate combinations. Consequently, some vigilance to avoid overfitting must be applied in the design processes of the GCF-RAB machines.

H. Classification load

A reasonable estimate of the computational effort which is needed to classify a new sample comes from considering the corresponding number of multiplications and, if Look-Up-Tables (LUTs) are not used to obtain the outputs of the nonlinear transformations, the number of such transformations. (Note that we do not distinguish between exponentials and sigmoids, which is in favour of standard RAB schemes).

With respect to the number of multiplications:

- To obtain the argument of each gate element, D multiplications are needed for the squared distance, and the result has to be multiplied by $1/2\sigma_n^2$. So, a total number of $N(D+1)$ multiplications are necessary for each ensemble step.
- To get each output of the gate, N additional multiplications are needed.
- An MLP with M hidden neurons requires MD (input layer) plus M (output layer) multiplications.
- For standard RAB schemes, one more multiplication (by α_t) is needed for each step. The same is true for GCF-RAB ensembles if we add gate element outputs before multiplying.

Thus, if T is the number of base learners, a RAB ensemble needs $T(M(D+1)+1)$ products, and a GCF-RAB ensemble requires $T(M(D+1)+N+1)+N(D+1)$ products.

Table III.A shows the average values corresponding to the designs we have obtained for the eight problems under analysis. Note that, if LUTs are used, those values indicate the classification computational load.

It is easy to see that standard RAB architectures require less multiplications to classify a new sample for problems Co and Cr, and also less than S1 for Ab and less than S2 and S2(R) for Io. In all the other cases, GCF-RAB ensembles need less multiplications than standard RAB (note that the best cases are shown in boldface). Thus, a computational advantage of the proposed designs is not unexpectable, but it seems to appear frequently.

If nonlinear transformation values are calculated, the computational load has an additional component which depends on the number of nonlinear elements included in each machine ensemble; i.e., $T(M+1)$ for RAB and $T(M+1)+N$ for GCF-RAB designs. Table III.B shows the corresponding values.

Although to provide a good comparison in this case requires to know an equivalent number of multiplications for each nonlinearity (in the practice, a relatively high number), for the problems and designs we are considering things are relatively clear. With the only possible doubt of S1 for Ab, GCF-RAB classifications are computationally less expensive than RAB for Ab, He, Kw, and Ri, and some of them also for Br and Io. The only problems that lead to less computationally demanding RAB architectures are Co and Cr (note that Cr is a curious case which originates designs of equal performance).

According to the above, it can be said that GCF-RAB ensembles offer the possibility of obtaining not only better performance, but even less classification effort in comparison with standard RAB schemes.

IV. Conclusions

This paper proposes a new approach to include a kernel type gate to combine learners' outputs in a Real AdaBoost

TABLE III
NUMBER OF MULTIPLICATIONS AND NUMBER OF NONLINEAR FUNCTIONS TO CLASSIFY A NEW PATTERN BY EACH ENSEMBLE

Problem	A. Number of multiplications					B. Number of nonlinear functions				
	RAB	GCF-RAB				RAB	GCF-RAB			
		S1	S2	S1(R)	S2(R)		S1	S2	S1(R)	S2(R)
Ab	1154.4	1574.7	856.0	**557.0**	882.2	156.0	98.9	98.5	**64.0**	94.0
Br	1299.3	807.9	882.0	**495.0**	2286.8	149.1	66.3	72.0	**58.6**	235.2
Co	**707.7**	5686.2	941.3	2543.1	1325.2	101.1	230.1	**97.1**	137.7	108.2
Cr	**188.7**	1093.1	314.8	519.2	812	**33.3**	82.3	35.3	64.9	101.5
He	7570.2	1975.6	**1630.8**	2118.7	2322.5	399.6	98.0	**96.5**	120.3	139.5
Io	3907.2	**1751.3**	5721.3	1788.9	6975.8	132.2	**61.9**	140.7	65.3	209.3
Kw	1347.8	**229.8**	427.0	501.6	629.1	468.8	**63.4**	94.0	148.0	185.5
Ri	4190.5	773.2	772.4	595.8	**452.1**	1416.1	**84.7**	189.2	174.4	106.2

ensemble construction, leading to the Gate Controlled Fusion Real AdaBoost ensembles. The advantages of this proposal has been experimentally checked for a series of different design procedures, obtaining systematically positive answers. This way, one more method to cope with Real AdaBoost intrinsic limitations has been made available. Additionally, the robustness of the corresponding designs, that need a computationally costly Cross-Validation process, has also been checked.

The price for increasing the already good performance of Real AdaBoost algorithms is an important increase in the design computational effort (up to several orders of magnitude), mainly due to Cross-Validation explorations. However, a detailed analysis has shows that, in many cases, the operation load (computational effort to classify new samples) of the proposed ensembles is lower than the load of conventional Real AdaBoost algorithms.

Besides the opportunities of improving the particular designs that are proposed here, their base —to combine strong aspects of Real AdaBoost and Mixtures of Experts— opens the door for investigating other principled procedures to combine local and global (weak) learning machines.

REFERENCES

[1] A. J. Sharkey, (ed.), *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*. London, UK: Springer-Verlag, 1999.

[2] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ: Wiley, 2004.

[3] L. Rokach, *Pattern Classification Using Ensemble Methods*. Singapore: World Scientific, 2010.

[4] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, pp. 197–227, 1990.

[5] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. of Computer and System Sciences*, vol. 55, pp. 119 – 139, 1997.

[6] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, pp. 297–336, 1999.

[7] H. Drucker, R. E. Schapire, and P. Simard, "Boosting performance in neural networks," *Intl. J. of Pattern Recognition and Artificial Intelligence*, vol. 7, pp. 705–719, 1993.

[8] Y. LeCun, L. D. Jackel, H. A. Eduard, N. Bottou, C. Cortes, J. S. Denker, H. Drucker, E. Sackinger, P. Simard, and V. Vapnik, "Learning algorithms for classification: A comparison on handwritten digit recognition," in *Neural Networks: The Statistical Mechanics Perspective*, J. H. Oh, C. Kwon, and S. Cho, (eds.). Singapore: World Scientific, 1995, pp. 261–276.

[9] H. Drucker and C. Cortes, "Boosting decision trees," in *Advances in Neural Information Proc. Sys. 8*, D. S. Touretzky, M. Mozer, and M. E. Hasselmo, (eds.). Cambridge, MA: MIT Press, 1996, pp. 479–485.

[10] H. Schwenk and Y. Bengio, "Adaboosting neural networks," in *Proc. 7th Intl. Conf. on Artificial Neural Networks* (LNCS 1327), W. Gerstner, A. Germond, M. Hasler, and J. D. Nicoud, (eds.). Berlin: Springer, 1997, pp. 967–972.

[11] L. Breiman, "Randomizing outputs to increase prediction accuracy," *Machine Learning*, vol. 40, pp. 229–242, 2000.

[12] D. Mease and A. Wyner, "Evidence contrary to the statistical view of boosting," *J. Machine Learning Res.*, vol. 9, pp. 131–156, 2008.

[13] R. E. Schapire, P. Bartlett, Y. Freund, and W. S. Lee, "Boosting the margin: a new explanation for the effectiveness of voting methods," *The Annals of Statistics*, vol. 26, pp. 1651–1686, 1998.

[14] L. Breiman, "Arcing classifiers," *The Annals of Statistics*, vol. 26, pp. 801–824, 1998.

[15] L. Breiman, "Prediction games and arcing algorithms," *Neural Computation*, vol. 11, pp. 1493–1517, 1999.

[16] J. R. Quinlan, "Boosting first-order learning," in *Proc. 7th Intl. Workshop Algorithmic Learning Theory* (LNCS 1160), S. Arikawa and A. Sharma, (eds.). Berlin: Springer, 1996, pp. 143–155.

[17] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, vol. 36, pp. 105–139, 1999.

[18] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, pp. 1895–1923, 1998.

[19] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine Learning*, vol. 40, pp. 139–157, 2000.

[20] Y. Freund, "An adaptive version of the boost by majority algorithm," *Machine Learning*, vol. 43, pp. 293–318, 2001.

[21] G. Rätsch, T. Onoda, and K. R. Müller, "Regularizing adaboost," in *Proc. Advances in Neural Information Proc. Sys. 11*, M. Kearns, S. Solla, and D. Cohn, (eds.). Cambridge, MA: MIT Press, 1999, pp. 564–570.

[22] G. Rätsch, T. Onoda, and K. R. Müller, "Soft margins for adaboost," *Machine Learning*, vol. 42, pp. 287–320, 2001.

[23] G. Rätsch and M. K. Warmuth, "Efficient margin maximizing with boosting," *J. Machine Learning Res.*, vol. 6, pp. 2131–2152, 2005.

[24] G. Lugosi and N. Vayatis, "On the Bayes-risk consistency of regularized boosting methods," *The Annals of Statistics*, vol. 32, pp. 30–35, 2004.

[25] P. L. Bartlett and M. Traskin, "Adaboost is consistent," *J. Machine Learning Res.*, vol. 8, pp. 2347–2368, 2007.

[26] Y. Sun, S. Todorovic, and J. Li, "Reducing the overfitting of adaboost by controlling its data distribution skewness," *Intl. J. Pattern Recognition and Artificial Intelligence*, vol. 20, pp. 1093–1116, 2006.

[27] C. Shen and H. Li, "Boosting through optimization of margin distributions," *IEEE Trans. Neural Networks*, vol. 21, pp. 659–666, 2010.

[28] 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.

[29] V. Gómez-Verdejo, J. Arenas-García, and A. R. Figueiras-Vidal, "A dynamically adjusted mixed emphasis method for building boosting ensembles," *IEEE Trans. Neural Networks*, vol. 19, pp. 3 –17, 2008.

[30] C.-X. Zhang, J.-S. Zhang, and G.-Y. Zhang, "An efficient modified boosting method for solving classification problems," *J. Computational and Applied Mathematics*, vol. 214, pp. 381 – 392, 2008.

[31] N. Duffy and D. P. Helmbold, "Leveraging for regression," in *Proc. 13th Annual Conf. on Computational Learning Theory*. San Francisco, CA: Morgan Kaufmann, 2000, pp. 208–219.

[32] G. Rätsch, S. Mika, and M. K. Warmuth, "On the convergence of leveraging," in *Advances in Neural Information Proc. Sys. 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, (eds.). Cambridge, MA: MIT Press, 2002, pp. 487–494.

[33] R. Meir and G. Rätsch, "An introduction to boosting and leveraging," in *Advanced Lectures on Machine Learning* (LNCS 2600), S. Mendelson and A. Smola, (eds.). New York, NY: Springer, 2003, pp. 118–184.

[34] L. Mason, P. L. Bartlett, and J. Baxter, "Improved generalization through explicit optimization of margins," *Machine Learning*, vol. 38, pp. 243–255, 2000.

[35] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Functional gradient techniques for combining hypotheses," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, (eds.). Cambridge, MA: MIT Press, 2000, pp. 221–246.

[36] S. Mannor and R. Meir, "On the existence of linear weak learners and applications to boosting," *Machine Learning*, vol. 48, pp. 219–251, 2002.

[37] J. Lu, K. Plataniotis, A. Venetsanopoulos, and S. Li, "Ensemble-based discriminant learning with boosting for face recognition," *IEEE Trans. Neural Networks*, vol. 17, pp. 166 –178, 2006.

[38] J. H. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *The Annals of Statistics*, vol. 28, pp. 337–407, 2000.

[39] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, pp. 367 – 378, 2002.

[40] H. Masnadi-Shirazi and N. Vasconcelos, "Cost-sensitive boosting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, pp. 294 –309, 2011.

[41] V. Boyarshinov and M. Magdon-Ismail, "Efficient optimal linear boosting of a pair of classifiers," *IEEE Trans. Neural Networks*, vol. 18, pp. 317 –328, 2007.

[42] C. Shen and H. Li, "On the dual formulation of boosting algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2216 –2231, 2010.

[43] S. Chen, H. He, and E. Garcia, "Ramoboost: Ranked minority oversampling in boosting," *IEEE Trans. Neural Networks*, vol. 21, pp. 1624 –1642, 2010.

[44] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "Semiboost: Boosting for semi-supervised learning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, pp. 2000–2014, 2009.

[45] K. Chen and S. Wang, "Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, pp. 129–143, 2011.

[46] S. Chen, X. Wang, X. Hong, and C. Harris, "Kernel classifier construction using orthogonal forward selection and boosting with fisher ratio class separability measure," *IEEE Trans. Neural Networks*, vol. 17, pp. 1652–1656, 2006.

[47] P. Sun and X. Yao, "Sparse approximation through boosting for learning large scale kernel machines," *IEEE Trans. Neural Networks*, vol. 21, pp. 883–894, 2010.

[48] N. García-Pedrajas, C. García-Osorio, and C. Fyfe, "Nonlinear boosting projections for ensemble construction," *J. Machine Learning Res.*, vol. 8, pp. 1–33, 2007.

[49] C.-X. Zhang and J.-S. Zhang, "Rotboost: A technique for combining rotation forest and adaboost," *Pattern Recognition Letters*, vol. 29, pp. 1524–1536, 2008.

[50] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.

[51] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, pp. 181–214, 1994.

[52] M. I. Jordan and L. Xu, "Convergence results for the EM approach to mixtures of experts architectures," *Neural Networks*, vol. 8, pp. 1409–1431, 1995.

[53] F. Peng, R. A. Jacobs, and M. A. Tanner, "Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition," *J. of the American Statistical Association*, vol. 91, pp. 953–960, 1996.

[54] R. A. Jacobs, F. Peng, and M. A. Tanner, "A Bayesian approach to model selection in hierarchical mixtures-of-experts architectures," *Neural Networks*, vol. 10, pp. 231–241, 1997.

[55] M. Olteanu and J. Rynkiewicz, "Estimating the number of components in a mixture of multilayer perceptrons," *Neurocomputing*, vol. 71, pp. 1321–1329, 2008.

[56] A. S. Weigend, M. Mangeas, and A. N. Srivastava, "Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting," *Intl. J. of Neural Sys.*, vol. 6, pp. 373–399, 1995.

[57] A. J. Zeevi, R. Meir, and R. J. Adler, "Time series prediction using mixtures of experts," in *Advances in Neural Information Proc. Sys. 9*, M. Mozer, M. I. Jordan, and T. Petsche, (eds.). Cambridge, MA: MIT Press, 1997, pp. 309–318.

[58] J. C. Principe, "Modeling, segmentation, and classification of nonlinear nonstationary time series," in *Nonlinear Dynamical Systems : Feedforward Neural Network Perspectives*, I. W. Sandberg, J. T. Lo, C. L. Fancourt, J. C. Principe, S. Katagiri, and S. Haykin, (eds.). New York, NY: Wiley, 2001, pp. 103–209.

[59] M. S. Mirian, M. N. Ahmadabadi, B. N. Araabi, and R. R. Siegwart, "Learning active fusion of multiple experts' decisions: An attention-based approach," *Neural Computation*, vol. 23, pp. 558–591, 2011.

[60] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[61] A. F. R. Rahman and M. C. Fairhurst, "A new hybrid approach in combining multiple experts to recognise handwritten numerals," *Pattern Recognition Letters*, vol. 18, pp. 781–790, 1997.

[62] J. B. Hampshire-II and A. Waibel, "The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 751–769, 1992.

[63] J. Fritsch, M. Finke, and A. Waibel, "Adaptively growing hierarchical mixtures of experts," in *Advances in Neural Information Proc. Sys. 9*, M. Mozer, M. I. Jordan, and T. Petsche, (eds.). Cambridge, MA: MIT Press, 1997, pp. 459–465.

[64] V. Ramamurti and J. Ghosh, "Structurally adaptive modular networks for nonstationary environments," *IEEE Trans. Neural Networks*, vol. 10, pp. 152–160, 1999.

[65] B. Shahbaba and N. Radford, "Nonlinear models using Dirichlet process mixtures," *J. Machine Learning Res.*, vol. 10, pp. 1829–1850, 2009.

[66] C. Wang, X. Liao, L. Carin, and D. B. Dunson, "Classification with incomplete data using Dirichlet process priors," *J. Machine Learning Res.*, vol. 11, pp. 3269–3311, 2010.

[67] A. Rao, D. Miller, K. Rose, and A. Gersho, "Mixture of experts regression modeling by deterministic annealing," *IEEE Trans. Signal Proc.*, vol. 45, pp. 2811 –2820, 1997.

[68] L. Lin, X. Wang, and D. Yeung, "Combining multiple classifiers based on a statistical method for handwritten Chinese character recognition," *Intl. J. Pattern Recognition and Artificial Intelligence*, vol. 19, pp. 1027–1040, 2005.

[69] M. H. Nguyen, H. A. Abbass, and R. I. McKay, "A novel mixture of experts model based on cooperative coevolution," *Neurocomputing*, vol. 70, pp. 155 – 163, 2006.

[70] Y. Ge and W. Jiang, "On consistency of Bayesian inference with mixtures of logistic regression," *Neural Computation*, vol. 18, pp. 224–243, 2006.

[71] S. R. Waterhouse and A. J. Robinson, "Non-linear prediction of acoustic vectors using hierarchical mixtures of experts," in *Advances in Neural Information Proc. Sys. 7*, G. Tesauro, D. S. Touretzky, and T. K. Leen, (eds.). Cambridge, MA: MIT Press, 1995, pp. 835–842.

[72] A. Carvalho and M. Tanner, "Mixtures-of-experts of autoregressive time series: Asymptotic normality and model specification," *IEEE Trans. Neural Networks*, vol. 16, pp. 39–56, 2005.

[73] L. Ohno-Machado and M. A. Musen, "Modular neural networks for medical prognosis: Quantifying the benefits of combining neural networks for survival prediction," *Connection Science*, vol. 9, pp. 71–86, 1997.

[74] R. Ebrahimpour, E. Kabir, H. Esteky, and M. R. Yousefi, "View-independent face recognition with mixture of experts," *Neurocomputing*, vol. 71, pp. 1103–1107, 2008.

[75] Z. Fu, A. Robles-Kelly, and J. Zhou, "Mixing linear SVMs for nonlinear classification," *IEEE Trans. Neural Networks*, vol. 21, pp. 1963–1975, 2010.

[76] A. Omari and A. R. Figueiras-Vidal, "Gate generated functional weights perceptron classifiers," to be submitted to the IEEE Trans. Neural Networks.

[77] R. Avnimelech and N. Intrator, "Boosted mixture of experts: An ensemble learning scheme," *Neural Computation*, vol. 11, pp. 483–497, 1999.

[78] G. Rätsch and M. K. Warmuth, "Maximizing the margin with boosting," in *Proc. 15th Annual Conf. on Computational Learning Theory*. London, UK: Springer-Verlag, 2002, pp. 334–350.

[79] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Local boosting of decision stumps for regression and classification problems," *J. of Computers*, vol. 1, pp. 30–37, 2006.

[80] M. Kawakita and S. Eguchi, "Boosting method for local learning in statistical pattern recognition," *Neural Computation*, vol. 20, pp. 2792–2838, 2008.

[81] Z. Chun-Xia and Z. Jiang-She, "A local boosting algorithm for solving classification problems," *Computational Statistics & Data Analysis*, vol. 52, pp. 1928–1941, 2008.

[82] R. Meir, R. El-Yaniv, and S. Ben-David, "Localized boosting," in *Proc. 13th Annual Conf. on Computational Learning Theory*. San Francisco, CA: Morgan Kaufmann, 2000, pp. 190–199.

[83] E. Mayhua-López, V. Gómez-Verdejo, and A. R. Figueiras-Vidal, "Improving boosting performance with a local combination of learners," in *Proc. Intl. Joint Conf. on Neural Networks*, Barcelona, 2010, pp. 1983–1990.

[84] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281–294, 1989.

[85] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 302–309, 1991.

[86] J. C. Platt, "Leaning by combining memorization and gradient descent," in *Advances in Neural Information Proc. Sys. 3*, R. P. Lippmann, J. E. Moody, and D. S. Touretzky, (eds.). San Mateo, CA: Morgan Kaufmann, 1991, pp. 714–720.

[87] B. Fritzke, "Supervised learning with growing cell structures," in *Advances in Neural Information Proc. Sys. 6*, J. D. Cowan, G. Tesauro, and J. Alspector, (eds.). San Mateo, CA: Morgan Kaufmann, 1994, pp. 255–262.

[88] S. Elanayar and Y. C. Shin, "Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems," *IEEE Trans. Neural Networks*, vol. 5, pp. 594 –603, 1994.

[89] S. Chen, "Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning," *Electronics Letters*, vol. 31, no. 2, pp. 117–118, 1995.

[90] J. Arenas-García, A. Figueiras-Vidal, and A. J. Sharkey, "The beneficial effects of using multi-net systems that focus on hard patterns," in *Multiple Classifier Systems* (LNCS 2709), T. Windeatt and F. Roli, (eds.). Berlin: Springer, 2003, pp. 45–54.

[91] E. I. Chang and R. P. Lippmann, "A boundary hunting radial basis function classifier which allocates centers constructively," in *Advances in Neural Information Proc. Sys. 5*, S. J. Hanson, J. D. Cowan, and C. L. Giles, (eds.). San Mateo, CA: Morgan Kaufmann, 1993, pp. 139–146.

[92] M. B. Almeida, A. P. Braga, and J. P. Braga, "SVM-KM: Speeding SVMs learning with a priori cluster selection and k-means," in *Proc. 6th Brazilian Symp. Neural Networks*, Rio de Janeiro, 2000, pp. 162–167.

[93] A. Lyhyaoui, M. Martínez-Ramón, I. Mora, M. Váquez, J.-L. Sancho, and A. R. Figueiras-Vidal, "Sample selection via clustering to construct support vector-like classifiers," *IEEE Trans. Neural Networks*, vol. 10, pp. 1474–1481, 1999.

[94] B. Schölkopf, S. Kah-Kay, C. J. C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Trans. Signal Proc.*, vol. 45, pp. 2758–2765, 1997.

[95] H. J. Shin and S. Cho, "Pattern selection for support vector classifiers," in *Proc. 3rd Intl. Conf. on Intelligent Data Engineering and Automated Learning*, Manchester, UK, 2002, pp. 469–474.

[96] H. J. Shin and S. Cho, "How many neighbors to consider in pattern pre-selection for support vector classifiers?" in *Proc. Intl. Joint Conf. on Neural Networks*, Portland, OR, 2003, pp. 565–570.

[97] H. J. Shin and S. Cho, "Fast pattern selection for support vector classifiers," in *Proc. 7th Pacific-Asia conf. Advances in Knowledge Discovery and Data Mining* (LNAI 2637), Seoul, Korea, 2003, pp. 376–387.

[98] H. J. Shin and S. Cho, "Neighborhood property-based pattern selection for support vector machines," *Neural Computation*, vol. 19, pp. 816–855, 2007.

[99] Y.-S. Hwang and S.-Y. Bang, "An efficient method to construct a radial basis function neural network classifier and its application to unconstrained handwritten digit recognition," in *Proc. Intl. Conf. on Pattern Recognition*, Vienna, Austria, 1996, pp. 640–644.

[100] J. T. Y. Kwok, "Moderating the outputs of support vector machine classifiers," *IEEE Trans. Neural Networks*, vol. 10, pp. 1018–1031, 1999.

[101] B. D. Ripley, "Neural networks and related methods for classification," *J. Royal Statistical Society*, vol. 56, pp. 409–456, 1994.

[102] A. Frank and A. Asuncion, "UCI machine learning repository," University of California, Irvine, School of Information and Computer Sciences, 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[103] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, UK: Cambridge Univ. Press, 1996.

Efraín Mayhua-López was born in Arequipa, Perú, in 1975. He received the B.S. Degree in Electronic Engineer from Universidad Nacional San Agustin de Arequipa, Arequipa, Perú, in 2000. He received the M.S. degree in E-Business: Telecommunications and New Business Models from Universidad de Cantabria, Cantabria, Spain, in 2005. He received the M.Sc. degree in Multimedia and Communications from Universidad Carlos III de Madrid, Madrid, Spain, in 2010. Currently, he is a Ph.D. Student in Multimedia and Communications and teaching assistant at the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Spain. His research interests include the fields of signal processing, machine learning, and their applications.

Vanessa Gómez-Verdejo was born in Madrid, Spain, in 1979. She received the Telecommunication Engineering degree in 2002 from Universidad Politécnica de Madrid. In 2007, she obtained the PhD from Universidad Carlos III de Madrid, where she is currently a Visiting Professor in the Department of Signal Theory and Communications. Her research interests are related to the machine learning algorithms, mainly neural network ensembles and boosting methods. She has coauthored around 20 papers, including journal and conference contributions. She has participated in several R+D projects with public funding and companies, what has provided her with an extensive experience in solving real-world problems.

Aníbal R. Figueiras-Vidal (S'74-M'76-SM'84) received the Telecommunication Engineer degree (honors) from Universidad Politécnica de Madrid, Madrid, Spain, in 1973, and the Doctor degree in Telecommunication Engineering (honors) from Universidad Politécnica de Barcelona, Barcelona, Spain, in 1976. Currently, he is a Professor of Signal Theory and Communications at Universidad Carlos III de Madrid. He has (co)authored more than 350 journal and conference papers in areas of his interests. His present research covers digital signal processing, neural networks, and learning theory. He is a Member of the Spain Royal Academy of Engineering. Dr. Figueiras-Vidal has received "Honoris Causa" doctorate degrees from Universidad de Vigo, Vigo, Spain, in 1999, and Universidad Católica San Pablo, Arequipa, Perú, in 2011.