Pattern Recognition Letters 88 (2017) 1–5

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Boosting ensembles with controlled emphasis intensity

Anas Ahachad, Lorena Álvarez-Pérez*, Aníbal R. Figueiras-Vidal

GAMMA-L+ / Department Signal Theory and Communications, University Carlos III of Madrid,Leganés (Madrid), 28911, Spain

ARTICLE INFO

Article history:
Received 18 July 2016
Available online 14 January 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Boosting
Classification
Emphasis

ABSTRACT

Boosting ensembles have deserved much attention because their high performance. But they are also sensitive to adverse conditions, such as noisy environments or the presence of outliers. A way to fight against their degradation is to modify the forms of the emphasis weighting which is applied to train each new learner. In this paper, we propose to use a general form for that emphasis function, which not only includes an error dependent and a proximity to the classification boundary dependent term, but also a constant value which serves to control how much emphasis is applied. Two convex combinations are used to consider these terms, and this makes possible to control their relative influence. Experimental results support the effectiveness of this general form of boosting emphasis.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Boosting is the most celebrated family of algorithms to build classifier ensembles. Its key idea is to iteratively train each learner paying more attention to examples that are difficult to classify by the previously available partial ensemble and, after it, to aggregate learner's output to that of the partial ensemble. Adaboost [10] and Real Adaboost (RA) [20] were its original forms. These designs minimize an exponential function of the margin product (target by output value) or an upper bound of it, respectively. Yet a huge number of modifications and extensions have appeared after them [19]. It is remarkable that the exact form of the emphasis –the example weighting factor for training– is not essential to get good results [5,6], although different forms can lead to better or worse performances in a problem-dependent manner.

One of the most valuable characteristics of boosting algorithms is that they oppose a serious resistance to overfitting [9,14,18,21]. But there are evidences of overfitting phenomena in some particular situations [7,8,15]. It was found that overfitting tends to appear when dealing with very noisy problems or when there are many outliers. It seems clear that to pay much attention to erroneous samples under these circumstances can increase these difficulties.

Several modifications have been introduced to deal with this drawback, for instance, [15,16,22,23]. Among these modifications, [11,12] proposed to combine the proximity to the classification boundary of each training example with an error measure in a parametric form. In this way, it is possible to balance the emphasis weighting among highly erroneous samples and examples that are close to the classification boundary, that have a great risk of becoming misclassified. Experimental results showed the effectiveness of this approach.

A second step is taken in [1,2], where the above mixed emphasis is also applied to the K nearest neighbors of each sample, and the overall weight for each sample is a convex combination of the individual and the average neighbor emphasis. A version in which the combination of the error and the proximity terms is selected for each learner according to the minimization of the edge parameter, which is called DWK-RA (Dynamic Weighting K-neighbour Real Adaboost), provides excellent experimental results. However, DWK-RA design requires a lot of computational effort, much of which is due to the cross validation of its many additional parameters: that of combining error and proximity (for each learner), that of combining individual and neighbor emphases, and the value of K, as well as the determination of the nearest neighbors for each sample.

In this contribution, we propose an alternative further step: Including a constant term with the combination of the error and proximity emphases. This will serve to graduate the intensity of that mixed emphasis, limiting the increased attention which is paid to the above mentioned types of examples, thus producing effects that are qualitatively similar to those of DWK-RA, but with a much lower training computational cost. This kind of emphasis has been successful in improving deep classifiers using auxiliary machines [3], allowing performance improvements much higher

* Corresponding author.
E-mail address: lalvarez@tsc.uc3m.es (L. Álvarez-Pérez).

http://dx.doi.org/10.1016/j.patrec.2017.01.009
0167-8655/© 2017 Elsevier B.V. All rights reserved.

Table 1
Characteristics of the benchmark problems.

Dataset	Notation	#Train C_1/C_{-1}	#Test C_1/C_{-1}	Dimension (D)
Abalone	Aba	2507 1238 / 1269	1670 843 / 827	8
Breast	Bre	420 145 / 275	279 96 / 183	9
Crabs	Cra	120 59 / 61	80 41 / 39	7
Credit	Cre	414 167 / 247	276 140 / 136	15
Diabetes	Dia	468 172 / 296	300 96 / 204	8
German	Ger	700 214 / 486	300 86 / 214	20
Hepatitis	Hep	93 70 / 23	62 53 / 9	19
Image	Ima	1300 736 / 564	1010 584 / 426	18
Ionosphere	Ion	201 101 / 100	150 124 / 26	34
Kwok	Kwo	500 300 / 200	10200 6120 / 4080	2
Ripley	Rip	250 125 / 125	1000 500 / 500	2
Waveform	Wav	400 124 / 276	4600 1523 / 3077	21

than simpler forms. However, let us remark from the beginning that there is not any theoretical guarantee of getting this advantage in all the practical situations: A relative overemphasis of examples that are near to the boundary can create even worse difficulties than overfitting, and the need of empirically the values of the emphasis parameters can lead to suboptimal designs.

The rest of the paper is structured as follows. In Section 2, we present and justify the emphasis function we propose. We will consider binary problems, although the formulation can be easily extended to multiclass situations. Section 3 presents some experiments and discusses their results in comparison with those of RA-type ensembles and a non-moderated version of the proposed emphasis. The main conclusions of our work close this contribution.

2. The proposed emphasis function

According to the above, we will consider the emphasis

$$p_m\left(\mathbf{x}^{(n)}\right) = \alpha + (1-\alpha)\left[\frac{\beta\left(t^{(n)} - o_{m-1}^{(n)}\right)^2}{4} + (1-\beta)\left(1 - o_{m-1}^{(n)\,2}\right)\right] \tag{1}$$

where p_m is the weight for the example $\{\mathbf{x}^{(n)}, t^{(n)}\}$ (observation vector and its target, ± 1) for training learner m, $o_{m-1}^{(n)}$ is the aggregated output of the previous $m-1$ learners for that example (aggregation is carried out according to its standard RA form), and α, β are non-trainable design parameters. Obviously, β is a convex combination parameter, $0 \le \beta \le 1$, which balances the contribution of the term corresponding to the error, $\left(t^{(n)} - o_{m-1}^{(n)}\right)^2$, and the term corresponding to the proximity to the boundary, $1 - o_{m-1}^{(n)\,2}$. On the other hand, we regulate the intensity of the resulting mixed emphasis with a constant term: Since a factor in the emphasis weights is irrelevant, we combine a constant term α with $1-\alpha$ times the convex combination of the error and proximity terms, in order to allow a simple exploration: $0 \le \alpha \le 1$. Note that $\alpha = 0$ will reduce the emphasis to a convex combination of error and proximity, β serving to balance their relative importance. This is equivalent to the mixed emphasis which was introduced in [11,12], but using alternative analytical measures for error and proximity. If we also take $\beta = 1$, we have a quadratic error cost form of a pure RA, which we will call Alternative RA (ARA). In general, α and β can be established by means of a Cross Validation (CV) process.

For the sake of clarity, let us insist: There are three components in (1). The first is the constant term α: When it takes high values, the intensity of emphasis is reduced, and this can be beneficial when solving some problems. The other two terms, that are combined with α in a convex manner, consider the error, $t^{(n)} - o^{(n)}$, which is measured in the classical quadratic form, and the proximity to the border, $1 - o^{(n)\,2}$, which leads to pay more attention to samples that give near to zero outputs in the auxiliary machine, i.e., to samples that are near the classification boundary; so, they are critical for the performance of the resulting classification ensemble. There is also a convex combination for these two terms.

With respect to the auxiliary machine, or guide, which provides the values of $o^{(n)}$ to be used in (1), there are evidences of the advantage of using relatively powerful machines offering outputs not very different from those expected with the emphasized design. Thus, using the partial ensemble which is available when training each learner is an appropriate selection: This partial ensemble will be good enough in the final steps of the building process, and the similarity is obvious.

Of course, many other error and proximity measures could be employed in (1), and results would be better or worse depending on the database under analysis. But we invoke [5] to defend that our objective is to check if moderating the emphasis with $\alpha \neq 0$ can be beneficial, and not to explore how different measures work in different problems. Note that the form of (1) is computationally efficient.

3. Experiments and their discussion

3.1. Databases

We will apply (1) for building boosting ensembles for 12 well-known databases that are frequently used as benchmark sets for this kind of experiments: Crabs and Ripley [17], Kwok [13], and the rest (Abalone, Breast, Credit, Diabetes, German, Hepatitis, Image, Ionosphere, and Waveform), from [4]. Table 1 presents their main characteristics. We will denote these databases by their three first letters from now here. We remark that the practical reason to select these databases is to allow direct comparisons with the results of the references that evaluated different emphasis forms, that used just the same databases.

3.2. Learners and training

We will use one hidden layer (weak) Multi-Layer Perceptrons (MLPs) as learners because they are unstable machines, and this makes them sensitive to differences in the emphasis function. They are trained by the Back-Propagation algorithm to minimize the weighted squared error between the desired output and what the network actually outputs, initializing all the weights at random values from a $[-0.2, 0.2]$ uniform distribution. The learning rate for both layers is set to be 0.01, which has been experimentally proven to be enough to reach convergence. The number of hidden units, H, is established by means of a 20-run $\times$ 5-fold CV, which also serves to determine the values of α and β, that are explored from 0 to 1 in steps of size 0.1. An 80/20 early stopping mode is applied to stop training.

The final results come from training the cross-validated designs 50 times.

3.3. Results and their discussion

We will present performance results for five types of designs:

- The proposed Controlled RA (CRA), where α and β are established by means of CV;
- Two algorithms that do not include the moderation mechanism, i.e., with $\alpha = 0$: β RA, which includes both error and proximity terms –we repeat that it is a mixed emphasis scheme analogous to those in [11,12],– and the above mentioned ARA, which corresponds to $\alpha = 0,\ \beta = 1$.
- And, finally and for completeness, two schemes that include the moderating constant α, but only one output-dependent emphasis term: That of error, α1RA ($\beta = 1$), and that of proximity, α0RA ($\beta = 0$).

Table 2 shows the corresponding experimental results, average error rates ± standard deviations in %, for 50 runs with the CV selected parameters in each case. These parameters also appear inside brackets (averages ± standard deviations for M, the number of learners). We will discuss these results in an ordered form.

* CRA vs. ARA: CRA performs better than ARA for 8 databases (Bre, Dia, Ger, Ima, Ion, Kwo, Rip, and Wav) and it is not worse than ARA for the other four, according to Wilcoxon tests with 95% confidence level.
  We remark that the Wilcoxon tests results must be considered as merely indicative, since we are not dealing with strictly independent experiments. It can be easily checked that, in our cases, these results are much similar to those offered by the simple rule-of-thumb of accepting performance differences when the difference between error averages is higher than the average of typical deviations, a rule-of-thumb we will apply here for other less relevant comparisons.
  The above results permit to conclude that CRA provides a clear advantage with respect to the standard ARA.
* β RA vs. ARA: β RA does not improves ARA for two databases that show advantage for CRA (Ion and Wav) according to the same Wilcoxon tests, and it improves ARA results for the other six. Thus, β RA can be considered better than the standard ARA, but not as good as CRA. We emphasize that this occurs even using a very simple CV mechanism, a fact that affects to CRA more than to β RA, because CRA includes two non-trainable parameters, α and β, and β RA only the second.
  Together, the above results mean that both the constant and the proximity to the boundary terms are useful to increase performance in enough number of cases, although most of the advantage usually comes from the second.
* α1RA and α0RA: By constraining β to each of its two extreme values, we get the effects that could be expected. When using α1RA –a moderated version of ARA–, things become clearly worse with respect to CRA for Cre and Rip. It is remarkable that there is not significant degradation for databases that used a high value of α in CRA (Dia, Ger, and Hep). On the other hand, when applying α0RA –a combination of a moderation and a proximity to the boundary terms–, there is more degradation for Cre, but none for Rip. This indicates that a moderated emphasis according to the proximity to the boundary can be occasionally better than a moderated version of the traditional emphasis according to the classification error, a new and interesting finding.

Table 2
Average error rate ± standard deviation (%) for the five boosting ensembles for the databases of the experiments. CV values for α, β and H (number of hidden units) are also included, as well as statistics (average ± standard deviation) of the number of machine elements of the ensembles, M. Symbol • indicates a significative difference with respect to ARA according to a Wilcoxon test with a significance level of 95%, for CRA and β RA.

	CRA (M) (H, α, β)	β RA (M) (H)	ARA $\alpha = 0$, $\beta = 1$ (M) (H)	α1RA $\beta = 1$ (M) (H, α)	α0RA $\beta = 0$ (M) (H, α)
Aba	19.2 ± 0.2 (18.4 ± 0.5) (5/0.4/0.1)	19.3 ± 0.3 (24.6 ± 3.3) (5/0.1)	19.2 ± 0.4 (28.7 ± 3.8) (8)	19.3 ± 0.4 (19.6 ± 1.2) (5/0.2)	19.3 ± 0.2 (19.2 ± 0.8) (5/0.3)
Bre	2.1 ± 0.2 • (16.9 ± 3.4) (2/0.1/0.1)	2.1 ± 0.4 • (29.0 ± 9.4) (6/0.5)	2.5 ± 0.4 (20.1 ± 6.2) (2)	2.1 ± 0.2 (13.6 ± 0.5) (5/0.5)	2.1 ± 0.2 (14.0 ± 0.3) (3/0.6)
Cra	2.5 ± 0 (98.0 ± 3.8) (2/0.1/0.1)	2.5 ± 0 (98.9 ± 3.7) (2/0.1)	2.5 ± 0 (89.0 ± 7.4) (2)	2.5 ± 0 (92.8 ± 4.8) (2/0.1)	2.5 ± 0 (99.2 ± 3.2) (2/0.1)
Cre	7.4 ± 1.3 (18.1 ± 1.5) (2/0/0.9)	7.4 ± 1.3 (18.1 ± 1.5) (2/0.9)	7.4 ± 1.3 (18.0 ± 1.8) (2)	8.7 ± 0.8 (15.9 ± 0.7) (2/0.1)	10.8 ± 0.8 (15.2 ± 0.6) (3/0.5)
Dia	22.2 ± 0.8 • (18.7 ± 0.5) (2/0.9/0.3)	22.5 ± 0.9 • (19.9 ± 0.8) (2/0.1)	25.8 ± 1.4 (35.9 ± 7.9) (6)	22.3 ± 0.8 (18.6 ± 0.6) (2/0.9)	22.1 ± 0.8 (18.7 ± 0.6) (2/0.4)
Ger	22.2 ± 1.1 • (17.5 ± 0.9) (3/0.8/0.2)	22.8 ± 1.3 • (20.5 ± 1.9) (2/0.1)	25.9 ± 1.5 (56.9 ± 12.3) (6)	22.4 ± 0.9 (18.6 ± 1.1) (3/0.7)	22.6 ± 1.0 (17.7 ± 0.7) (2/0.8)
Hep	6.9 ± 0.8 (19.7 ± 2.1) (9/0.9/0.6)	7.2 ± 1.2 (22.2 ± 3.9) (18/1)	7.2 ± 1.2 (22.2 ± 3.9) (18)	7.1 ± 0.9 (23.0 ± 2.6) (18/0.1)	7.0 ± 0.9 (25.9 ± 3.8) (18/0.8)
Ima	3.0 ± 0.4 • (60.9 ± 9.0) (15/0.2/0.2)	3.1 ± 0.3 • (64.0 ± 9.0) (15/0.1)	4.1 ± 0.6 (32.5 ± 3.4) (15)	3.3 ± 0.5 (16.7 ± 0.6) (15/1)	3.1 ± 0.4 (54.1 ± 10.3) (15/0.4)
Ion	3.9 ± 0.7 • (17.1 ± 1.5) (2/0.6/0)	4.3 ± 0.9 (18.2 ± 1.4) (2/0.3)	4.3 ± 0.8 (29.8 ± 6.2) (7)	4.2 ± 0.8 (18.0 ± 1.7) (7/1)	3.9 ± 0.7 (17.1 ± 1.5) (2/0.6)
Kwo	11.6 ± 0.1 • (22.6 ± 4.5) (15/0/0.5)	11.6 ± 0.1 • (22.6 ± 4.5) (15/0.5)	11.8 ± 0.2 (18.7 ± 1.5) (6)	11.7 ± 0.1 (16.5 ± 0.5) (13/0.9)	11.7 ± 0.1 (16.9 ± 0.7) (13/0.7)
Rip	9.0 ± 0.2 • (31.6 ± 5.3) (14/0.3/0.1)	9.1 ± 0.2 • (37.7 ± 6.3) (15/0.1)	9.3 ± 0.3 (37.1 ± 6.4) (43)	9.6 ± 0.3 (22.7 ± 3.2) (48/0.1)	9.0 ± 0.2 (35.5 ± 5.8) (13/0.1)
Wav	11.3 ± 0.3 • (32.2 ± 10.0) (4/0.2/0.1)	11.6 ± 0.5 (17.2 ± 3.2) (2/0.2)	11.5 ± 0.3 (40.1 ± 6.3) (6)	11.3 ± 0.3 (16.0 ± 1.2) (3/0.7)	11.3 ± 0.4 (24.4 ± 6.0) (3/0.3)

Table 3
Average error rate ± standard deviation (%) for standard RA and ARA for the databases of the experiments. CV values for α, β and H (number of hidden units) are also included, as well as statistics (average ± standard deviation) of the number of learners of the ensembles, M.

	RA	ARA
Aba (M) (H)	19.4 ± 0.02 (31.2 ± 0.4) (4)	19.2 ± 0.4 (28.7 ± 3.8) (8)
Bre (M) (H)	2.6 ± 0.4 (21.3 ± 4.2) (6)	2.5 ± 0.4 (20.1 ± 6.2) (2)
Cra (M) (H)	2.5 ± 0 (11.1 ± 0.8) (2)	2.5 ± 0 (89.0 ± 7.4) (2)
Cre (M) (H)	10.1 ± 0.7 (29.4 ± 6.5) (2)	7.4 ± 1.3 (18.0 ± 1.8) (2)
Dia (M) (H)	20.6 ± 0.8 (33.2 ± 5.0) (2)	25.8 ± 1.4 (35.9 ± 7.9) (6)
Ger (M) (H)	22.3 ± 0.7 (33.6 ± 5.9) (2)	25.9 ± 1.5 (56.9 ± 12.3) (6)
Hep (M) (H)	8.9 ± 1.8 (22.2 ± 3.9) (17)	7.2 ± 1.2 (22.2 ± 3.9) (18)
Ima (M) (H)	3.0 ± 0.4 (21.2 ± 3.1) (11)	4.1 ± 0.6 (32.5 ± 3.4) (15)
Ion (M) (H)	4.9 ± 0.9 (13.4 ± 4.5) (5)	4.3 ± 0.8 (29.8 ± 6.2) (7)
Kwo (M) (H)	11.7 ± 0.01 (29.3 ± 0.1) (15)	11.8 ± 0.2 (18.7 ± 1.5) (6)
Rip (M) (H)	9.7 ± 0.01 (28.9 ± 0.9) (48)	9.3 ± 0.3 (37.1 ± 6.4) (43)
Wav (M) (H)	11.7 ± 0.4 (30.1 ± 6.1) (2)	11.5 ± 0.3 (40.1 ± 6.3) (6)

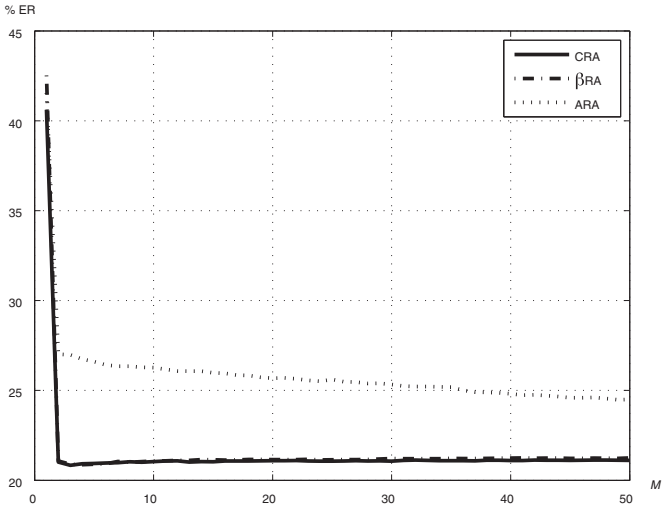


Fig. 1. Building average error rate (%), % ER, curves vs. number of learners, M, for CRA ($\alpha = 0.9,\ \beta = 0.3$) (continuous line), βRA ($\beta = 0.1$) (dotdash line) and ARA (dotted line) in Dia.

3.4. Some deeper perspectives

To appreciate the differences of results for different emphasis methods, Table 3 shows the performance of ARA and standard RA (the latter are taken from [2], Table 2). It appears that the winner method is problem-dependent: ARA for Cre, Hep, Ion, and Rip, and RA for Dia, Ger, and Ima.

Comparing the results of CRA with the best method in [2], DWK-RA, there are also clear differences for Cre (advantage of CRA) and for Ger, Hep, and Ima (advantage of DWK-RA) (we insist in the higher computational load that designing DWK-RA ensembles requires, as discussed in the fourth paragraph of the Introduction). These differences could be attributed to the different proximity and error measures that are used in both families of methods. But it must be noticed that DW (Dynamic Weighting) techniques select the proximity and hybrid emphasis terms for each learner, by maximizing the edge value –see [11]–, and this is an implicit advantage, and it also eliminates the need of cross-validating the combination parameter.

To check if this simplification of the CV processes has any influence in the results, it is useful to visualize the convergence of the algorithms we are introducing. Figs. 1 and 2 present the training and test error evolution with the number of learners in Dia for the algorithms we are introducing in this paper with their optimal CV parameterization: CRA with $\alpha = 0.9,\ \beta = 0.3,\ \beta$RA with $\beta = 0.1$, and ARA. Notice that both CRA and βRA reach (approx.) their best training performance for 2 learners, and they keep this performance when the number of learners increases. This will provoke stopping problems, and, consequently, difficulties in selecting their parameters by CV. In fact, an omniscient design of CRA (i.e., by selecting its non-trainable parameters according to the test results) leads to $\alpha = 0.3,\ \beta = 0.6$ with the same value of H ($H = 2$) and similar numbers of learners (19.3 ± 0.5)... But with a much better performance: 19.1 ± 0.4%. It is true that the omniscient design is invalid, but it serves to establish the potential limits of any CV process. Thus, it is obvious that improved stopping and CV processes will further increase the quality of the proposed designs, although the computational load will also increase. However, given the performance results of the different techniques we are exploring here, it can be concluded that CRA is a very good boosting algorithm which needs a relatively moderate computational effort for its design. This means that introducing an emphasis control parameter, such as α, is an efficient and effective possibility to build boosting ensembles.

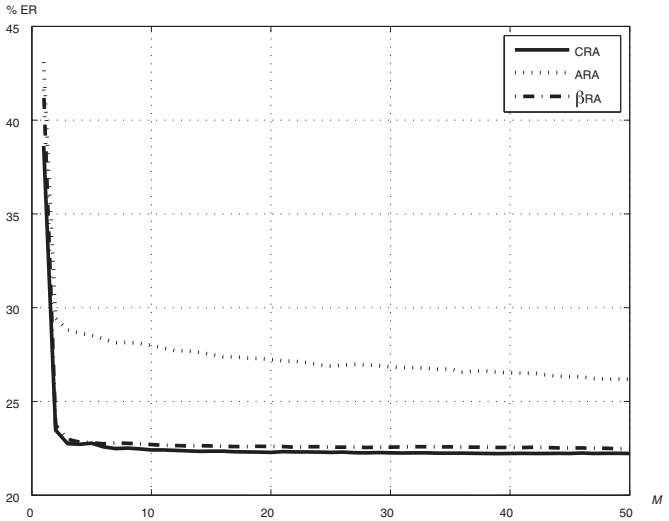


Fig. 2. Test average error rate (%), % ER, curves vs. number of learners, M, for CRA ($\alpha = 0.9,\ \beta = 0.3$) (continuous line), βRA ($\beta = 0.1$) (dotdash line) and ARA (dotted line) in Dia.

4. Conclusions

Including a constant term into the emphasis weights to construct boosting ensembles can be useful to obtain better performance. Experimental results for a number of benchmark databases support that this option is an effective possibility: Obtaining advantage requires not more than an easy and computationally affordable cross-validation process.

Further steps along this research direction will be dedicated to analyzing the sensitivity of the performances with respect to α and β and, subsequently, establishing complementary rules for improving these designs.

Acknowledgments

This work has been partly supported by research grants CASI-CAM-CM (S2013/ICE-2845,DGUI-CM), and Macro-ADOBE (TEC2015-67719-P, MINECO).

The authors thank the anonymous reviewers, who helped a lot to significantly improve the first version of this contribution.

References

[1] A. Ahachad, A. Omari, A.R. Figueiras-Vidal, Smoothed emphasis for boosting ensembles, in: I. Rojas, G. Joya, J. Cabestany (Eds.), Advances in Computational Intelligence (LNCS 7902), Springer, Berlin, 2013, pp. 367–375.

[2] A. Ahachad, A. Omari, A.R. Figueiras-Vidal, Neighborhood guided smoothed emphasis for Real Adaboost ensembles, Neural Proc. Letters 42 (2015) 155–165.

[3] R.F. Alvear-Sandoval, A.R. Figueiras-Vidal, An experiment in pre-emphasizing diversified deep neural classifiers, in: Proceedings of European Symposium on Artificial Neural Networks, 23, 2016, pp. 527–532. Bruges (Belgium)

[4] K. Bache, M. Lichman, UCI Machine learning repository, school of information & computer sciences, Univ. California at Irvine, 2013. http://archive.ics.uci.edu/m)

[5] L. Breiman, Arcing classifiers, Ann. Statistics. 26 (1998) 801–839.

[6] L. Breiman, Combining predictors, in: A.J.C. Sharkey (Ed.), Combining Artificial Neural Nets Ensemble and Modular Multi-net Systems, Springer, London, 1999, pp. 31–50.

[7] L. Breiman, Prediction games and arcing algorithms, Neural Comput. 11 (7) (1999) 1493–1517.

[8] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput. 10 (7) (1998) 1895–1923.

[9] H. Drucker, C. Cortes, L.D. Jackel, Y. LeCun, V. Vapnik, Boosting and other ensemble methods, Neural Comput. 6 (6) (1994) 1289–1301.

[10] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1997) 119–139.

[11] V. Gómez-Verdejo, J. Arenas-García, A.R. Figueiras-Vidal, A dynamically adjusted mixed emphasis method for building boosting ensembles, IEEE Trans. Neural Netw. 19 (1) (2008) 3–17.

[12] V. Gómez-Verdejo, M. Ortega-Moral, J. Arenas-García, A.R. Figueiras-Vidal, Boosting by weighting critical and erroneous samples, Neurocomput. 69 (7–9) (2006) 679–685.

[13] J.T.Y. Kwok, Moderating the outputs of support vector machine classifiers, IEEE Trans. Neural Netw. 10 (5) (1999) 1018–1031.

[14] Y. LeCun, L.D. Jackel, H.A. Eduard, N. Bottou, C. Cartes, J.S. Denker, H. Drucker, E. Sackinger, P. Simard, V. Vapnik, Learning algorithms for classification: a comparison on handwritten digit recognition, in: J. Oh, C. Kwon, S. Cho (Eds.), Neural networks, World Scientific, 1995, pp. 261–276.

[15] L. Mason, P.L. Bartlett, J. Baxter, Improved generalization through explicit optimization of margins, Mach. Learn. 38 (3) (2000) 243–255.

[16] G. Rätsch, M.K. Warmuth, Efficient margin maximizing with boosting, J. Mach. Learn. Res. 6 (2005) 2131–2152.

[17] B.D. Ripley, N.L. Hjort, Pattern Recognition and Neural Networks (1st ed.), Cambridge University Press, New York, NY, USA, 1995.

[18] R. Schapire, Y. Singer, P. Barlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Statistics. 26 (1998) 1651–1686.

[19] R.E. Schapire, Y. Freund, Boosting: foundations and algorithms, 3rd, MIT Press, Cambridge, MA, 2012.

[20] R.E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Mach. Learn. 37 (1999) 297–336.

[21] H. Schwenk, Y. Bengio, Adaboosting neural networks: application to on-line character recognition, in: W. Gerstner, A. Germond, M. Hasler, J.-D. Nicoud (Eds.), Proceedings of Intlernational Conference on Artificial Neural Networks (ICANN'97), Springer Berlin Heidelberg, 1997, pp. 967–972.

[22] Y. Sun, S. Todorovic, J. Li, Reducing the overfitting of adaboost by controlling its data distribution skewness., Int. J. Pattern Recognit. Artif. Intell. 20 (7) (2006) 1093–1116.

[23] C.-X. Zhang, J.-S. Zhang, G.-Y. Zhang, An efficient modified boosting method for solving classification problems, J. Comput. Appl. Math. 214 (2) (2008) 381–392.