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.

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/

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.