Robust Asymmetric Adaboost^{*}

Pablo Ormeño¹, Felipe Ramírez¹,
Carlos Valle¹, Héctor Allende-Cid¹, and Héctor Allende^{1,2}

¹ Departamento de Informática, Universidad Técnica Federico Santa María
Avenida España 1680, Valparaíso, Chile

² Facultad de Ingeniería y Ciencia, Universidad Adolfo Ibáñez
Avenida Padre Hurtado 750, Viña del Mar, Chile

Abstract. In real world pattern recognition problems, such as computer-assisted medical diagnosis, events of a given phenomena are usually found in minority, making it necessary to build algorithms that emphasize the effect of one of the classes at training time. In this paper we propose a variation of the well-known Adaboost algorithm that is able to improve its performance by using an asymmetric and robust cost function. We assess the performance of the proposed method on two medical datasets and synthetic datasets with different levels of imbalance and compare our results against three state-of-the-art ensemble learning approaches, achieving better and comparable results.

Keywords: ensemble learning, adaboost, asymmetric cost functions, robust methods.

1 Introduction

Over the past years machine learning methods for automatic pattern recognition have obtained successful results in a variety of real-world applications. Learning from data has proved to be very useful for aiding decision making on matters such as planning, forecasting, visualization, process control, etc. In some domains, however, traditional pattern recognition algorithms fail to provide adequate solutions given the difficulties inherent to some phenomena. One of such difficulties is the asymmetry in the distribution of certain observable events, as occurs in computer-assisted diagnosis of rare diseases.

This problem has been recently studied and a considerable number of algorithms have been proposed to provide more suitable solutions given its particular features [7,6]. Two major trends of pattern recognition methods have made substantial contributions in the field: cost-sensitive learning and ensemble methods.

One of the most popular ensemble-based techniques is Adaboost [3], a sequential iterative approach that fits an additive model through a forward stage-wise

^{*} This work was supported by the following Research Grants: Fondecyt 1110854 and FB0821 Centro Científico Tecnológico de Valparaíso. Partial support was also received from CONICYT (Chile) Ph.D. Grant 21080414.

approach [5], by training a single predictor to minimize the residual of the previous ensemble in each round. In other words, this algorithm combines a set of weak classifiers weighting them according to their individual performances in the training phase. Thus, the weight of wrongly classified samples increases and the weight of correctly classified samples decreases.

Since its introduction, several variations of Adaboost have been proposed to solve different not previously-considered issues [4,10]. One of the major problems of the original version of Adaboost is that it is unable to handle the presence of outlying observations in the data, which causes an excessive increase of their sampling weights as they are most likely misclassified in each iteration. The authors of [1] proposed an improvement to Adaboost to address this flaw. The algorithm is termed Robust Alternating Adaboost (Radaboost) because it operates by literally alternating between Adaboost and its inverse variation [8]. The authors report that the proposed method outperforms the original version when the data has outlier samples [11].

However, none of these techniques are appropriate in pattern recognition problems where class distributions are imbalanced or the misclassification costs are not equivalent. Often, these problems appear simultaneously, as in the aforementioned example, where observations of ill patients are heavily outnumbered by the observations of healthy ones, and it is more important to detect the disease rather than the absence of it. In other words, the cost of false negatives (also called *type-II errors*) is much greater than that of false positives (also called *type-I errors*). Hence, a different treatment for each kind of error is needed, which often means to penalize more one than the other.

In this work we propose an algorithm that incorporates robustness and asymmetry allowing to solve the problem of imbalance by differentiating the kinds of errors committed. The remainder of the paper is organized as follows: in Section 2 we give a brief literature review on the methods that inspired this work, in Section 3 we describe in detail our proposal, in Section 4 we perform an experimental comparative study with the proposed method and the previously described related literature approaches, and finally in Section 5 we conclude and discuss possible further work.

2 Boosting Algorithms

In this section we show details of the algorithms that inspired our work and whose understanding is required to proceed to introduce our improvement. We begin by introducing the traditional form of the Adaboost algorithm with its inverse variation, then we explain briefly the Radaboost algorithm, and finally we cite a cost-sensitive boosting algorithm.

The Adaboost algorithm [3] takes as input a training set $\mathcal{Z} = \{(x_1, y_1) \dots (x_n, y_n)\}$ where each x_i is a variable that belongs to $\mathcal{X} \subset \mathbb{R}^d$ and each y_i is some label in $\mathcal{Y}$. In this particular paper we assume that $\mathcal{Y} = \{-1, 1\}$. Adaboost calls a weak or base learning algorithm repeatedly in a sequence of stages $t = 1 \dots T$.

The main idea of AdaBoost is to maintain a sampling distribution over the training set, which is used to train the learner at each round t . Let $\mathcal{D}_t(i)$ be

the sampling weight assigned to example i on round t . At the beginning of the algorithm the distribution is uniform, *i.e.* $\mathcal{D}_1(i) = \frac{1}{n} \forall i$, and at each round, the weights of the misclassified examples are increased, so that the following base learner is forced to focus on the *hard* examples of the training set. The job of each learner is to find an appropriate hypothesis $h_t : \mathcal{X} \rightarrow \{-1, 1\}$ for the distribution $\mathcal{D}_t$. The goodness of the obtained hypothesis can be quantified as the weighted error $\epsilon_t = \Pr_{i \sim \mathcal{D}_t}[h_t(x_i) \neq y_i] = \sum_{i: h_t(x_i) \neq y_i} \mathcal{D}_t(i)$. Note that the error is measured with respect to the distribution $\mathcal{D}_t$ on which the weak learner was trained.

Once the weak hypothesis h_t is obtained, its relevance is computed $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$ and the sampling distribution is updated according to the rule $\mathcal{D}_{t+1}(i) = \frac{1}{Z_t} \mathcal{D}_t(i) e^{-\alpha_t y_i h_t(x_i)}$, where Z_t is a normalization factor. This rule has the aforementioned effect on the sampling weights of correctly classified and misclassified objects. After a sequence of T rounds have been performed, the final hypothesis H is computed as

$$H_T(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right) \quad (1)$$

The Inverse Adaboost algorithm is a variation of the original form that operates with an inverse distribution updating rule $\mathcal{D}_{t+1}(i) = \frac{1}{Z_t} \mathcal{D}_t(i) e^{\alpha_t y_i h_t(x_i)}$. This rule basically underweights misclassified observations and vice versa, contrary to the original approach. Thus, the base learners are forced to focus more on the *easy* examples of the training set.

The Radaboost algorithm [1] was proposed to address the problem of data with outliers. It operates by alternating between the classic and the inverse versions of Adaboost to create a more stable algorithm, which limits the influence of the empiric distribution by detecting and decreasing the weight of problematic examples to achieve better classification overall.

The authors introduce an alternative way to compute α_t to prevent excessive increases in sampling weights of difficult examples. As the sampling distribution is updated in every stage, after a few iterations the weight of misclassified samples will be considerably higher than that of the rest. To avoid this, two variables are introduced: the inverse variable $\beta(i)$ and the epoch variable $\text{age}(i)$ for each example $i = 1, \dots, n$ in the training set. If $\beta(i)$ is equal to 1, it means that the original Adaboost is being used to update the weights. Otherwise, if $\beta(i)$ is equal to -1, the inverse approach is being used. The $\text{age}(i)$ variable counts the number of times that the samples was sequentially misclassified. If this value is higher than a given threshold τ , then $\beta(i)$ takes the value -1 and alternates.

As mentioned earlier, one of the limitations of Adaboost is that it focuses on minimizing the overall classification error not taking into account the difference between the costs of the two possible types of error, given a certain problem. To solve this issue, Asymmetric Adaboost [12] was proposed, which works by introducing a non-uniform initial sampling distribution according to the asymmetry in the misclassification costs of the problem.

3 Robust Asymmetric Adaboost

As described in the latter section, Radaboost only focuses on solving the problem of outlying examples and does not provide special treatment for cases with asymmetric misclassification costs. In this section we propose a considerable number of modifications for the Radaboost algorithm in order to provide it with cost-sensitiveness in the context of medical applications. Our proposal consists in a cost-sensitive updating function that allows the algorithm to find solutions more suitable for a given problem considering its particular cost asymmetry.

We first define a non-uniform initial sampling distribution using a k_i parameter, which measures how many times the majority class outnumbers the minority class for samples i that belong to the minority class. Thus, the values of the positive examples are increased, allowing the algorithm to focus in minimizing the false negative rate.

$$D_1(i) = \frac{k_i}{\sum_i^m k_i}, \forall i = 1, \dots, m. \quad (2)$$

Note that the parameter k_i characterizes the level of imbalance in the class distribution of the training.

Then, based on the original Radaboost algorithm, we modify the limit for α_t taking the r th root, with $r \in N$, to values of ϵ_t lower than a given γ :

$$\alpha_t = \begin{cases} \frac{1}{2} \sqrt[r]{\ln(\frac{1-\epsilon_t}{\epsilon_t})} + \alpha_\gamma & \epsilon_t < \gamma \\ \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t}) & \epsilon_t \geq \gamma \end{cases} \quad (3)$$

where $\alpha_\gamma = \frac{1}{2} \ln(\frac{1-\gamma}{\gamma}) - \frac{1}{2} \sqrt[r]{\ln(\frac{1-\gamma}{\gamma})}$ is a constant needed to maintain the continuity of equation (3). This prevents the sampling weights of problematic examples to grow excessively in one iteration.

We also define a cost-sensitive pseudo-loss function ϵ , which takes into account that the cost of false negatives is greater than that of false positives in the context of medical applications. An important thing to note is that the new value of ϵ that is introduced in the equation (5), maintains its continuity.

Finally, we propose a new updating rule for the sampling distribution that takes into account the level of imbalance in the class distribution of the training set. In each iteration the distribution is computed for every example i as follows

$$D_{t+1}(i) = \frac{\exp(-\alpha_t y_i \sum_t h_t(x_i)) \exp(y_i \log \sqrt{k(i)})}{Z_t}, \quad (4)$$

where Z_t is a normalization factor. This rule overweights positive examples by means of the k_i parameter previously used to define the initial distribution. As a result, more minority class objects will be sampled for training and, in turn, learners of further iterations will be fed with balanced training data.

All aforementioned modifications to the Radaboost method can be seen in Algorithm 1. In this particular study, we used as base classifier the Support Vector Machine (SVM) [9].

Algorithm 1. Robust Asymmetric Adaboost

-
- 1: Given the training set $Z = \{(x_1, y_1), \dots, (x_n, y_n)\}$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$; define σ_{ini} , σ_{min} , and σ_{dec}
 - 2: Initialize the parameters: choose the k value, the threshold of epochs τ , the threshold of the error γ , σ with σ_{ini} , the robust parameter r , the sample distribution $D_1(i) = \frac{k_i}{\sum_{i=1}^n k_i}$, $\forall i = 1, \dots, n$, the inverse variable $\beta(i) = 1$ and the variable $age(i) = 0$ for each data in the sample (x_i, y_i) , $i = 1, \dots, n$.
 - 3: **for** $t=1, \dots, T$ **do**
 - 4: **while** $(\sigma > \sigma_{min})$ **do**
 - 5: Take the weak hypothesis $h_t : X \rightarrow \{-1, +1\}$
 - 6: Compute the error of the base classifier

$$\epsilon_t = \sum_i D_t(i) \frac{(1 - h_t(x_t)y_t)}{2} \quad (5)$$

- 7: **if** $\epsilon_t > 0.5$ **then**
 - 8: Decrease the value of σ , in σ_{dec} and go to step 5.
 - 9: **end if**
 - 10: Calculate α_t with the equation (3)
 - 11: Update the weights distribution as in equation (4)
 - 12: **if** The sample (x_i, y_i) is correctly classified by $H_T(x)$ ($H_T(x_i)y_t > 0$) **then**
 - 13: $age(i) = 0$ and $\beta(i) = 1$
 - 14: **else**
 - 15: Increase $age(i)$ in one.
 - 16: **if** $age(i) > \tau$ **then**
 - 17: $\beta(i) = -1$ and $age(i) = 0$
 - 18: **end if**
 - 19: **end if**
 - 20: **end while**
 - 21: **end for**
 - 22: Calculate the final strong hypothesis as in equation (1)
-

4 Experimental Results

In this section we present the results obtained with our proposal (*Robust Asymmetric* for short) and the other three related literature approaches: the classic Adaboost algorithm, the Radaboost algorithm (*Robust* for short) and the Asymmetric Adaboost algorithm (*Asymmetric* for short). We used the *Breast Cancer* and *Liver Disease* datasets from the UCI Machine Learning Repository [2], along with a synthetic dataset to assess the performance of the methods with different imbalance rates.

Each experiment was run 20 times to report the mean values and the variability of the performance measures. An 80% of the data was used for training and the remaining 20% for testing. The *Breast Cancer* dataset, has a total of 568 examples, where 212 correspond to the positive class (malignant tumor) and 356 to the negative class (benign tumor). The *Liver Disease* dataset has 345 examples, 145 of which are positive (presence of the disease) and 200 are negative.

The γ and r parameters were chosen empirically. The values of γ were set to 0.05, 0.1 and 0.2. The robustness parameter r was set to 2, 3 and 5. The k value, as stated earlier, is the rate of the number of examples from the majority class over the number of examples of the minority. For the *Breast Cancer* data set the value was $k = 1.69$ and for the *Liver Disease* data set was $k = 1.37$. The values σ and C , for the SVM with gaussian kernel were tuned using the methodology proposed in [9]. The performance measures used to compare the 4 models are the classification error and sensitivity over the testing set, where the latter is a performance measure related to the false negative type of error.

Figures 1 and 2 show the two performance measures as a function of the number of boosting iterations with the *Breast Cancer* and *Liver Disease* datasets, respectively.

In the *Breast Cancer* case we can clearly see that our approach outperforms all other methods in both measures, and has the lower variability as well. Meanwhile, in the *Liver Disease* case, we can see that our method has comparable classification error mean and variability, and it has the higher sensitivity mean, but was outperformed by the *Asymmetric* approach in variability.

Note that the sensitivity has a direct relation with the type-II error: the higher its value, the lower the probability of committing false negatives will be, which is often required in medical applications such as automated diagnosis. Hence, given the above results, our method is the most appropriate for these kinds of problem.

We also synthetically generated instances of the *Breast Cancer* dataset to evaluate the performance of the methods against the imbalance ratio of the data. The proportions tested were 10-90%, 30-70% and 50-50%. Table 1 shows the results of the experiments with the three synthetic instances at 20 boosting

Fig. 1. Performance measures of *Breast Cancer* data set as a function of the number of boosting iterations

Fig. 2. Performance measures of *Liver Disease* data set as a function of the number of boosting iterations

iterations. We can see that our proposal outperforms the other three methods in classification error for all the cases, however, in sensitivity our results are comparable with those of the *Asymmetric* approach.

Table 1. Results with synthetically imbalanced data

Proportion	Algorithm	Error	Sensitivity
10 - 90	Adaboost	0.0326 ± 0.0100	0.8801 ± 0.0689
	Asymmetric	0.0217 ± 0.0141	0.9452 ± 0.0774
	Robust	$0.0253 \pm \mathbf{0.0073}$	0.9037 ± 0.0581
	Robust Asymmetric	$\mathbf{0.0127} \pm 0.0103$	$\mathbf{0.9489} \pm \mathbf{0.0481}$
30 - 70	Adaboost	0.0357 ± 0.0168	0.9246 ± 0.0497
	Asymmetric	0.0374 ± 0.0160	$\mathbf{0.9304} \pm 0.0729$
	Robust	0.0357 ± 0.0137	$0.9239 \pm \mathbf{0.0388}$
	Robust Asymmetric	$\mathbf{0.0272} \pm \mathbf{0.0113}$	$\mathbf{0.9304} \pm 0.0397$
50 - 50	Adaboost	0.0391 ± 0.0178	$\mathbf{0.9754} \pm \mathbf{0.0209}$
	Asymmetric	$0.0408 \pm \mathbf{0.0094}$	0.9647 ± 0.0295
	Robust	$0.0408 \pm \mathbf{0.0094}$	0.9550 ± 0.0328
	Robust Asymmetric	$\mathbf{0.0238} \pm 0.0097$	0.9647 ± 0.0295

5 Conclusion and Future Work

In this paper we have proposed a cost-sensitive variation of the Radaboost algorithm that aims not only to improve the performance of the positive class, but to preserve the robustness of its original form against outlying observations. We conducted an experimental study and compared our technique with three

related state-of-the-art approaches over real and synthetic datasets. The results show that our method is at least comparable with literature approaches in sensitivity, and outperforms them in overall accuracy. It also proved to be robust against the class imbalance rate achieving a steady low error in comparison to the growing error of the other approaches as the imbalance increases.

We conclude that our method successfully solves the data asymmetry issue, providing a solid alternative to address problems of this nature. Future work may consider a study of the convergence of the algorithm to improve its computational performance by using simpler base learners without hindering its classification accuracy.

References

1. Allende-Cid, H., Salas, R., Allende, H., Nanculef, R.: Robust Alternating Adaboost. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 427–436. Springer, Heidelberg (2007)
2. Frank, A., Asuncion, A.: UCI machine learning repository (2010), <http://archive.ics.uci.edu/ml>
3. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting (1997)
4. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. *Annals of Statistics* 28, 2000 (1998)
5. Hastie, T., Tibshirani, R., Friedman, J.: *The elements of statistical learning*. Springer (2003)
6. Japkowicz, N., Myers, C., Gluck, M.: A novelty detection approach to classification. In: *Proceedings of the Fourteenth Joint Conference on Artificial Intelligence*, pp. 518–523 (1995)
7. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. *Intell. Data Anal.* 6, 429–449 (2002)
8. Kuncheva, L.I., Whitaker, C.J.: Using Diversity with Three Variants of Boosting: Aggressive, Conservative, and Inverse. In: Roli, F., Kittler, J. (eds.) MCS 2002. LNCS, vol. 2364, pp. 81–90. Springer, Heidelberg (2002)
9. Li, X., Wang, L., Sung, E.: Adaboost with SVM-based component classifiers. *Eng. Appl. Artif. Intell.* 21(5), 785–795 (2008)
10. Nock, R., Nielsen, F.: A generalization of discrete adaboost. *Artificial Intelligence* 171(1), 25–41 (2007)
11. Takenouchi, T., Eguchi, S.: Robustifying adaboost by adding the naive error rate. *Neural Computation* 16(4), 767–787 (2004)
12. Wang, Z., Fang, C., Ding, X.: Asymmetric real adaboost. In: *ICPR 2008*, pp. 1–4 (2008)