PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 15/07

Generalized Boosting Learning

Julio Cesar Duarte
Ruy Luiz Milidiú

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900
RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 15/07 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena July, 2007

Generalized Boosting Learning

Julio Cesar Duarte, Ruy Luiz Milidiú

{jduarte, milidiu}@inf.puc-rio.br

Abstract. Boosting is a machine learning technique to combine several *weak* algorithms and improve their accuracies. In each iteration, the algorithm changes the weights of the examples building a different classifier. A simple final voting scheme among each classifier defines the classification of a new instance. The most common used algorithm based on boosting is AdaBoost, which starts with an uniform distribution for the examples. Unfortunately, there is no guarantee that this is the best choice for a initial distribution. We propose a boosting approach to model this issue, in which one can set any initial weight distribution. Besides that, we can also introduce a cost function to charge errors in most representative examples. For instance, if examples come in a sequential order, with human intervention, the earlier learned examples are more representative. In this kind of environment, one can use a skewed initial distribution like Zipf or geometric. We show the necessary changes in the original algorithm to accommodate the choice of any initial weight distribution.

Keywords: Machine Learning, Ensemble Methods, Boosting.

Resumo. Boosting é uma técnica de aprendizado de máquina que combina diversos algoritmos *fracos* e melhora os seus desempenhos. Em cada iteração, o algoritmo altera os pesos dos exemplos construindo um diferente classificador. Um esquema simples de votação final entre os classificadores define a classificação de uma nova instância. O algoritmo baseado em Boosting mais comumente utilizado é o AdaBoost, que inicia com uma distribuição inicial para os exemplos. Infelizmente, não existe nenhuma garantia de que essa escolha seja a melhor para uma distribuição inicial. Nessa monografia, é proposta uma abordagem baseada em Boosting para modelar essa questão, onde pode se definir qualquer distribuição inicial para os pesos. Além disso, pode-se também introduzir uma função de custo que penaliza erros em exemplos mais representativos. Por exemplo, se exemplos são construídos de uma forma seqüencial, com intervenção humana, os exemplos anteriormente aprendidos são mais representativos. Nesse tipo de ambiente, pode ser utilizada uma distribuição inicial enviesada como Zipf ou Geométrica. Nessa monografia, são apresentadas as alterações necessárias ao algoritmo original para acomodar a escolha de uma distribuição inicial qualquer.

Palavras-chave: Aprendizado de Máquina, Métodos de Comitê, Boosting.

In charge of publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

1 Introduction

In modern information retrieval, text classification has become a great tool to help the process of great amount of texts. Normally, in text classification, one or various labels are applied to a document. For instance, if we want to typify which kind of messages a client e-mail is acquiring, we can split them in two types, Spam and Non-Spam. Usually, it is difficult to come up with complex high accuracy classifiers. On the other hand, it is a lot easier to come up with simple rules that has better accuracy than random guess.

Ensemble learning algorithms, like bagging[Breiman, 1996] or boosting[Freund, 1990, Schapire, 1990], are machine learning approaches that combine different machine learning algorithms or different views of the same algorithm to build a better classifier.

Boosting is normally used in combination with "weak" classifiers to increase its accuracy. At each iteration of boosting, a classifier is built by using a weighted version of the original corpus. To achieve better accuracies by using boosting, the used baseline algorithm must have high variance, or instability, with respect to its training corpus.

For instance, good examples of "weak" classifiers are: classifiers based on the frequencies of words and one-level decision trees.

There are several implementations that uses the boosting approach such as LPBoost [Demiriz et al., 2002], TotalBoost[Freund and Schapire, 1996], and the most popular one, AdaBoost[Freund and Schapire, 1995].

AdaBoost is an implementation of a boosting approach which uses an initial uniform distribution for the examples. We here, propose a generalization for AdaBoost, called AdaBoost.S., where we can choose whichever distribution we want for the examples. We show that our generalization has the same advantages of the original AdaBoost and that we can evaluate each iteration weight distribution in a similar way.

Using a non-uniform weight distribution can bring advantages, specially when using examples obtained by an Active Learning[Cohn et al., 1994] process, or when dealing with problems involving Rare Events[Weiss, 2005].

2 Boosting with a generalized example distribution

Let us assume that we are given an example set $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in X$ and $y_i \in \{-1, +1\}\}$. Let us also assume that this example set is not a random sample of the original distribution of (x, y). Usually, the examples are not randomly chosen, and have an initial weight distribution w.

There are two options we should consider. One is to reduce the training error on the example set by changing the initial distribution D_1 of the AdaBoost algorithm. The other is to introduce a relative cost function p that charges more to errors on more representative examples, or less representative classes. There is an interplay between this two options that we explain in section 3.

We can express the weighting function D_1 as

$$D_1(i) \ = \ Kw(i)$$

where $i = 1, ..., n$ and K is the normalizing constant expressed by

$$K \ = \ \frac{1}{\sum_{i=1}^n w(i)}$$

Although it is not required, it would be interesting to choose D_1 as a non increasing density function. Some parametric choices are: uniform, geometric and Zipf. In Table 1, we summarize the weighting established by these three densities.

Density	D_1	w
uniform	$1/n$	1
geometric	$q^{i-1}(1-q)/1-q^n$	q^i
Zipf	$1/H_n i$	$1/i$

Table 1: Weighting Densities

Now, we introduce *Adaboost.A*, a variant of the AdaBoost algorithm. In Algorithm 1, we show a pseudocode for *Adaboost.A*, which is very similar to the original Adaboost algorithm.

Algorithm 1 The boosting algorithm AdaBoost.A

Input: $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in X$ and $y_i \in \{-1, +1\}\}$, $w_{i=1}^n$ and $p_{i=1}^n$
Initialize $D_1(i) = K.w(i)$
for $i = 1$ **to** T **do**
  train base learner using distribution D_t
  get base classifier $h_t : X \to \{-1, +1\}$
  choose $\alpha_t \in \Re$
  update the example distribution
    $D_{t+1}(i) = D_t(i)e^{-\alpha_t y_i h_t(x_i)}/Z_t$
    where $Z_t = \sum_{i=1}^n D_t(i)e^{-\alpha_t y_i h_t(x_i)}$
end for
Output: the final classifier
  $H(x) = sign\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$

The novelty is that *Adaboost.A* accepts a general distribution for D_1 and a general cost function p that measures example errors. Due to this change, a different value of α_t is also required in order to guarantee that the error rate of the combined predictor H is improved.

Using a greedy strategy, we show in section 3 that with the value of α_t given by

$$\alpha_t = \frac{1}{2} ln \left(\frac{\sum_{i \in C_t} D_t(i)p(i)/w(i)}{\sum_{i \in M_t} D_t(i)p(i)/w(i)} \right) \tag{1}$$

we improve the training error of the final H, where $C_t = \{i | h_t(i) = y_i\}$ and $M_t = \{i | h_t(i) \neq y_i\}$

It is interesting to note that when D_1 is from the *Zipf family* then α_t is given by

$$\alpha_t = \frac{1}{2} ln \left(\frac{\sum_{i \in C_t} iD_t(i)p(i)}{\sum_{i \in M_t} iD_t(i)p(i)} \right) \tag{2}$$

Similarly, when D_1 and p are both *uniform* we get the same value as in standard AdaBoost. In this case, AdaBoost.A reduces to AdaBoost.

Finally, we observe that equation (1) is general, since it makes no extra assumption on the initial distribution D_1.

3 Training Error

Now, we should proceed to the evaluation of the boosting α_t parameter.

Let the final classifier H be given by

$$H(x) = sign\left(f(x)\right)$$

where the score $f(x)$ is defined by

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x).$$

Hence,

$$f(x) = H(x)|f(x)|.$$

The training error τ of the binary predictor H is usually defined by

$$\tau = \frac{|M|}{n}$$

where M is the mistake given by $M = \{i|H(i) \neq y_i\}$.

Whenever p is a *relative cost* function over the example set, one can define the weighted training error τ_p of the binary predictor H by

$$\tau_p = \sum_{i \in M} p(i).$$

Now, we generalize a well known bound on the training error τ_p.

First, observe that

$$\begin{aligned}
\sum_{i=1}^{n} e^{-y_i f(x_i)} p(i) &= \sum_{i=1}^{n} e^{-y_i H(x_i)|f(x_i)|} p(i) \\
&= \sum_{i \in M} e^{|f(x_i)|} p(i) + \sum_{i \notin M} e^{-|f(x_i)|} p(i) \\
&\geq \sum_{i \in M} 1.p(i) + \sum_{i \notin M} 0.p(i) \\
&= \sum_{i \in M} p(i) = \tau_p.
\end{aligned}$$

Therefore,

$$\tau_p \leq \sum_{i=1}^{n} e^{-y_i f(x_i)} p(i). \tag{3}$$

On the other hand,

$$\begin{aligned}
\sum_{i=1}^{n} e^{-y_i f(x_i)} p(i) &= \sum_{i=1}^{n} p(i) \prod_{t=1}^{T} e^{-y_i \alpha_t h_t(x_i)} \\
&= \sum_{i=1}^{n} p(i) \prod_{t=1}^{T} Z_t \frac{D_{t+1}(i)}{D_t(i)} \\
&= \left(\prod_{t=1}^{T} Z_t \right) \sum_{i=1}^{n} D_{T+1}(i) \frac{p(i)}{D_1(i)}
\end{aligned}$$

However,

$$\begin{aligned}\sum_{i=1}^{n} e^{-y_i f(x_i)} p(i) &= \frac{\sum_{i=1}^{n} e^{-y_i f(x_i)} p(i)}{\sum_{i=1}^{n} p(i)} \\ &= \frac{\sum_{i=1}^{n} e^{-y_i f(x_i)} p(i)}{\sum_{i=1}^{n} D_1(i) p(i)/D_1(i)}\end{aligned}$$

and so,

$$\begin{aligned}\sum_{i=1}^{n} e^{-y_i f(x_i)} p(i) &= \frac{\left(\prod_{t=1}^{T} Z_t\right) \sum_{i=1}^{n} D_{T+1}(i) p(i)/D_1(i)}{\sum_{i=1}^{n} D_1(i) p(i)/D_1(i)} \\ &= \prod_{t=1}^{T} \frac{\sum_{i=1}^{n} Z_t D_{t+1}(i) p(i)/D_1(i)}{\sum_{i=1}^{n} D_t(i) p(i)/D_1(i)} \\ &= \prod_{t=1}^{T} \left(\frac{e^{-\alpha_t} \sum_{i \in C_t} D_t(i) p(i)/D_1(i)}{\sum_{i=1}^{n} D_t(i) p(i)/D_1(i)} + \frac{e^{\alpha_t} \sum_{i \in M_t} D_t(i) p(i)/D_1(i)}{\sum_{i=1}^{n} D_t(i) p(i)/D_1(i)}\right)\end{aligned}$$

we can also rewrite this equation as follows,

$$\sum_{i=1}^{n} e^{-y_i f(x_i)} . p(i) = \prod_{t=1}^{T} R_t \tag{4}$$

where $C_t = \{i | h_t(i) = y_i\}$, $M_t = \{i | h_t(i) \neq y_i\}$ and

$$R_t = \frac{e^{-\alpha_t} \sum_{i \in C_t}^{n} D_t(i) p(i)/D_1(i)}{\sum_{i=1}^{n} D_t(i) p(i)/D_1(i)} + \frac{e^{\alpha_t} \sum_{i \in M_t}^{n} D_t(i) p(i)/D_1(i)}{\sum_{i=1}^{n} D_t(i) p(i)/D_1(i)}$$

Combining (3) and (4), we get that

$$\tau_p \leq \prod_{t=1}^{T} R_t. \tag{5}$$

and if we define A_t such as

$$A_t = \frac{\sum_{i \in M_t} D_t(i) p(i)/w(i)}{\sum_{i=1}^{n} D_t(i) p(i)/w(i)} \leq 1.$$

we can rewrite R_t as follows,

$$R_t = e^{-\alpha_t}(1 - A_t) + e^{\alpha_t}(A_t)$$

On each round t, we greedily choose α_t to minimize R_t, obtaining:

$$\frac{dR_t}{d\alpha_t} = -e^{-\alpha_t}(1 - A_t) + e^{\alpha_t} A_t = 0$$

$$\alpha_t = \frac{1}{2} ln \left(\frac{1 - A_t}{A_t}\right)$$

And for this value,

$$\begin{aligned} R_t &= \sqrt{\frac{A_t}{1-A_t}}(1-A_t) + \sqrt{\frac{1-A_t}{A_t}}(A_t) \\ &= 2\sqrt{A_t(1-A_t)} \leq 1 \end{aligned}$$

Whenever A_t is smaller than $1/2$, R_t reduces the bound on the training error τ_p. In that case, $\alpha_t \neq 0$ and h_t is included in the score function f.

4 Conclusions

The use of ensemble methods like boosting improves the accuracy of several machine learning algorithms. These methods create a series of weak classifiers that perform well for different kinds of examples. A simple voting method, based on each classifier's accuracy, is used to combine all classifiers.

The major contribution of this work is a generalization of the AdaBoost algorithm called AdaBoost.A, which can accept any initial weight distribution and an error cost function for the examples.

These changes in the original algorithm may provide great benefits when working with corpus with non-iid data or high unbalanced distributions. An example of the first case are the examples obtained by an Active Learning Tagging process with a minimum confidence choice. An example of the second case are the examples inside a corpus with rare classes.

References

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

[Cohn et al., 1994] Cohn, D. A., Atlas, L., and Ladner, R. E. (1994). Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

[Demiriz et al., 2002] Demiriz, A., Bennett, K. P., and Shawe-Taylor, J. (2002). Linear programming boosting via column generation. *Machine Learning*, 46(1-3):225–254.

[Freund, 1990] Freund (1990). Boosting a weak learning algorithm by majority. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*.

[Freund and Schapire, 1995] Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37.

[Freund and Schapire, 1996] Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156.

[Guo and Viktor, 2004] Guo, H. and Viktor, H. L. (2004). Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *SIGKDD Explor. Newsl.*, 6(1):30–39.

[Kim and Kim, 2004] Kim, H. and Kim, J. (2004). Combining active learning and boosting for naïve bayes text classifiers. *Advances in Web-Age Information Management*, pages 519–527.

[Kubat and Matwin, 1997] Kubat, M. and Matwin, S. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In *Proc. 14th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann.

[Meir and Rätsch, 2003] Meir, R. and Rätsch, G. (2003). An introduction to boosting and leveraging. *Advanced lectures on machine learning*, pages 118–183.

[Schapire, 2001] Schapire, R. (2001). The boosting approach to machine learning: An overview.

[Schapire, 1990] Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5:197–227.

[Schapire and Singer, 2000] Schapire, R. E. and Singer, Y. (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

[Weiss, 2005] Weiss, G. M. (2005). Mining with rare cases. In *The Data Mining and Knowledge Discovery Handbook*, pages 765–776. Springer.