ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Improving BAS Committee Performance with a Semi-Supervised Approach

Ruy Luiz Milidiú[1] and Julio Cesar Duarte[2]

1- Pontifícia Universidade Católica - Departamento de Informática
Marquês de São Vicente, 225, Gávea, Rio de Janeiro, RJ - Brazil

2- Centro Tecnológico do Exército - Divisão de Tecnologia da Informação
Américas, 28.705, Guaratiba, Rio de Janeiro, RJ - Brazil

Abstract. Semi-supervised Learning is a machine learning approach that, by making use of both labeled and unlabeled data for training, can significantly improve learning accuracy. Boosting is a machine learning technique that combines several *weak* classifiers to improve the overall accuracy. At each iteration, the algorithm changes the weights of the examples and builds an additional classifier. A well known algorithm based on boosting is AdaBoost, which uses an initial uniform distribution. Boosting At Start (BAS) is a boosting framework that generalizes AdaBoost by allowing any initial weight distribution and a cost function. Here, we present a scheme that allows the use of unlabeled data in the BAS framework. We examine the performance of the proposed scheme in some datasets commonly used in semi-supervised approaches. Our empirical findings indicate that BAS can improve the accuracy of the generated classifiers by taking advantage of unlabeled data.

1 Introduction

Supervised learning is a machine learning approach for learning a function from labeled data. The trainer algorithm receives a set of labeled data which is called the train set and is used to find a good model adjustment.

Usually, the labeled data is divided into train and test set. This data set splitting is necessary to assure that the model is generalizing and not just memorizing the labeled data. Good performance on the test set is a good indication that the model would perform well on new data.

Unfortunately, the generation of labeled data is very expensive and usually depends on the skills of a human agent to manually tag the examples.

Semi-supervised learning, on the other hand, is a machine learning approach that makes use of both labeled and unlabeled data for training.

The main goal of semi-supervised learning is to take advantage of massive inexpensive unlabeled samples that are a by-product of ordinary enterprise processes. Based on this large sample set, it is not hard to infer several statistical properties of the domain that can be very helpful on designing efficient training schemes. Two basic semi-supervised learning approaches are Self-training and Co-training.

Self-training [1] goes in rounds. On the first round, we train the model using the train set. At each one of the next rounds, samples are first submitted to the current trained model. The model generates a classification for each one of the samples. The samples with the highest measure of confidence are included into the training set, assuming their estimated classification tags as the true ones. Using this enlarged train set, the model is trained again and updated.

Co-training [2] is very similar to Self-training. The main difference is that two models are simultaneously trained and they exchange high confidence samples with each other. One model gets the new examples from the other model.

Ensemble learning algorithms, like bagging [3] and boosting [4, 5], 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 a *weak* machine learning algorithm to increase its accuracy. At each boosting iteration, a classifier is built by using a new weighted version of the original corpus.

AdaBoost is a boosting implementation that uses an initial uniform distribution for the example weights. Alternatively, Boosting At Start (BAS) [6] is an AdaBoost generalization where we can choose any initial weight distribution for the examples. Through the BAS Committee algorithm, one can use BAS in order to generate a better classifier by taking advantage of the use of a non-uniform initial distribution for the examples.

Here, we propose a way to incorporate unlabeled data in the BAS Committee algorithm. In our experiments, we observed that our approach can improve the accuracy of the classifiers generated with only labeled data. We also show that these results are competitive with other state-of-the-art machine learning techniques. This is a strong evidence that the BAS Committee algorithm can take benefits of unlabeled data in the initial distribution determination. Therefore, BAS is a boosting algorithm that can provide better performance in a semi-supervised approach.

2 The BAS Algorithm

The *BAS* algorithm is a variant of AdaBoost, which accepts any initial weight distribution. The training process occurs similar to AdaBoost. In each iteration t a different classifier h_t is training based on a weighted D_t version of the training data (x, y). The main difference here is that, since *BAS* accepts a general initial distribution (D_1) based on a example weight function w, a different value of α_t (classifier's vote power) is necessary in order to guarantee that the error rate of the combined boosting classifier is improved. Also, the weight update process is changed since it relies on this modified α_t value.

It can be proved [6] that the new value of α_t is given by

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

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

In Algorithm 1, we show a pseudocode for the *BAS* algorithm.

Algorithm 1 The BAS algorithm.

```
Input: example set $\{(x_i, y_i)\}_{i=1}^{n}$ where $x_i \in X$ and $y_i \in \{-1, +1\}\}$
       example weight function $w$
       number of iterations $T$
Initialize $D_1(i) = K.w(i)$, where K is a normalization constant
for $t = 1$ to $T$ do
    Train base learner using distribution $D_t$
    Get base classifier $h_t : X \rightarrow \{-1, +1\}$
    Let $C_t = \{i|h_t(i) = y_i\}$ and $M_t = \{i|h_t(i) \neq y_i\}$
    Evaluate $\alpha_t = \frac{1}{2} ln \left( \frac{\sum_{i \in C_t} D_t(i)/w(i)}{\sum_{i \in M_t} D_t(i)/w(i)} \right)$
    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 BAS classifier
        $H(x) = sign \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right)$
```

3 Semi-supervised BAS Committee

In order to take advantage of any weight initialization, BAS Committee uses feature clustering to determine the initial weights for the examples in the BAS framework. Here, we propose a semi-supervised version of BAS Committee in which the unlabeled data is used in conjunction with the labeled data in order to help data clustering, thus helping the initial weight determination process.

In this scheme, the labeled data is initially, split into two datasets: a train set and a validation set.

In order to determine the initial weights, the unlabeled data is combined to the train set. This merged set is then separated into k clusters based on their common features. Possible clustering algorithms that can be used are *K-Means* [7], *Growing Neural Gas* [8] or any other unsupervised clustering algorithm. This process basically determines that if two examples are in a same cluster, they will have the same initial weight, otherwise, they will have different initial weights.

Then, an available distribution family is chosen to be used as the initial weights. The choice can derive from any known distribution.

Next, a permutation of this distribution is chosen and the weights are applied to the clusters found in the clustering step. The idea behind this, is to "disturb" the algorithm by assigning different weights to examples that are found in different clusters and equal weights to examples found in a same cluster.

Finally, the generated BAS classifier is applied to the validation set and its performance is evaluated.

This process is repeated over N iterations and the N' best BAS members are selected to form a committee, with equal voting power each.

The Semi-supervised BAS Committee is illustrated in Algorithm 2.

Algorithm 2 The Semi-supervised BAS Committee algorithm.

```
Input: labeled data set $LDS$
       unlabeled data set $UDS$
       clustering algorithm $CA$
       number of clusters $k$
       number of distribution families $f$
       weight distribution families $W_f$
       number of trained BAS Classifiers $N$
       BAS Committee size $N'$
Split $LDS$ into two subsets, a training set $Tr$ and a validation set $Va$
Apply $CA$ to $Tr + UDS$ obtaining the $k$-sized cluster data $Cl$
for $t = 0$ to $N - 1$ do
    Shuffle $W_f$ and apply to $Tr$ based on $Cl$, obtaining weight function $w$
    Train BAS Classifier $B_t$ using $Tr$ and $w$
    Evaluate the $B_t$ performance over $Va$
end for
Output: Committee formed with best $N'$ BAS Classifiers
        based on performance over $Va$
```

4 Experiments

In order to examine the performance of the Semi-supervised BAS Committee, we evaluate the proposed scheme using 13 artificial and real word two-class datasets [9] commonly cited in several works related to semi-supervised learning and boosting. Table 1 shows a brief description of these data sets.

	Abbr.	# Sets	# Features	Train Size	Test Size
Banana	ban	100	2	400	4900
Breast-Cancer	bca	100	9	200	77
Diabetes	dia	100	8	468	300
Flare-Solar	fls	100	9	666	400
German	ger	100	20	700	300
Heart	hea	100	13	170	100
Image	ima	20	18	1300	1010
Ringnorm	rin	100	20	400	7000
Splice	spl	20	60	1000	2175
Titanic	tit	100	3	150	2051
Thyroid	thy	100	5	140	75
Twonorm	two	100	20	400	7000
Waveform	wav	100	21	400	4600

Table 1: Thirteen datasets used in the experiments.

We conducted experiments comparing the proposed Semi-supervised BAS Committee (SSBASC) algorithm, the base learner system (BLS) used by the boosting members, AdaBoost and the best result reported in [9] which compares several state-of-the-art classifiers.

The base learner system used, except for the *banana* instance, is a Decision Stump, also implemented in WEKA 3 [10], which finds the best split for the features of the train set based on entropy. For the special case of *banana*, which has only two features, the Decision Stump has poor performance and is replaced by a C4.5 [11] Decision Tree derived from Quinlan's implementation which accepts weights and can generate limited-level trees.

The Semi-supervised BAS Committee algorithm uses the best 3 members chosen among 7 BAS classifiers, with 20% of the labeled data used for validation. K-means is the algorithm used to determine the best 5-cluster from the merged labeled and unlabeled data.

Although it is not required, it would be interesting to choose the initial weights as a non increasing density function. With this function, examples that come from different clusters have different levels of importance in the initial boosting iterations and are "filtered" together. Some parametric choices are arithmetic, geometric and Zipf. In this particular experimental setup, the initial weights can be set to any value from an arithmetic progression $(1, 2, 3, 4, 5)$, and, at each time, a different permutation is applied.

We also evaluated the experiments with different distributions and combinations for the classifiers of the final BAS Committee but, due to space restrictions, we show only the results for the best setup.

For each dataset, we perform train and test in all corresponding sets and report both the mean and standard deviation of the classification accuracy evaluated in the test sets. The reported results in Table 2 are for the final generated classifier. Bold values indicate the best figure for the problem.

	BLS	**AdaBoost**	**Best Reported** [9]		**SSBASC**
ban	84.44 ± 1.48	84.44 ± 1.48	KFD	**89.20 ± 0.50**	87.07 ± 0.54
bca	70.34 ± 3.52	73.42 ± 3.89	KFD	74.20 ± 4.60	**76.96 ± 3.83**
dia	71.90 ± 1.82	75.05 ± 1.42	KFD	76.80 ± 1.60	**77.34 ± 1.19**
fls	56.12 ± 1.82	67.61 ± 1.54	SVM	67.60 ± 1.80	**67.74 ± 1.61**
ger	70.18 ± 1.72	75.31 ± 2.13	SVM	76.40 ± 2.10	**76.49 ± 1.59**
hea	72.96 ± 3.00	80.55 ± 3.14	SVM	84.00 ± 3.30	**85.15 ± 2.68**
ima	87.67 ± 1.67	92.49 ± 2.31	ABR	**97.30 ± 0.60**	96.33 ± 0.45
rin	61.23 ± 0.97	91.82 ± 0.70	KFD	**98.50 ± 0.10**	92.17 ± 0.59
spl	77.00 ± 1.39	93.32 ± 0.41	ABR	90.50 ± 0.70	**93.73 ± 0.33**
tit	77.32 ± 2.82	76.75 ± 2.66	SVM	76.40 ± 1.00	**77.86 ± 1.73**
thy	78.00 ± 1.24	93.89 ± 1.28	KFD	95.80 ± 2.10	**96.21 ± 0.56**
two	66.47 ± 1.20	92.46 ± 0.56	KFD	**97.40 ± 0.20**	94.16 ± 0.37
wav	74.54 ± 2.37	86.72 ± 0.56	ABR	**90.20 ± 0.80**	87.36 ± 0.37

Table 2: Classification performance for all datasets.

As we can see in Table 2, the SSBASC approach consistently outperforms AdaBoost in all datasets and, in seven out of thirteen datasets, it also outperforms the best reported result. This is a great result since we are comparing SSBASC with not only one machine learning solution but the best out of five possibilities: Radial Basis Function (RBF), AdaBoost+RBF (AB), Regularized AdaBoost (ABR), Support Vector Machines (SVM) and Kernel Fischer Discriminant (KFD). For more details on the algorithms can be found in [9].

5 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 kind of examples.

Semi-supervised is a powerful machine learning technique which uses inexpensive data in order to improve a classifier's accuracy.

The major contribution of this work is a semi-supervised approach which can be applied to the BAS Committee algorithm and uses the unlabeled data in order to determine a better initial distribution for the BAS members.

This impact can be seen as a new way to introduce problem knowledge that comes from unlabeled data into boosting modeling. We provide evidences of this by reporting on experiments that explore this open avenue.

A next step in this work is to use the inherent knowledge from unlabeled data to also help the training of the base learner system. This is feasible by the use of Assemble [12], a Semi-supervised AdaBoost extension, and can be easily achieved since the BAS framework also provides a way to deal with a relative cost function that charges more or less to errors on different kinds of examples.

References

[1] Vincent Ng and Claire Cardie. Weakly supervised natural language learning without redundant views. In *Proceedings of the 2003 Conference of the North American Chapter of the ACL on Human Language Technology*, pages 94–101, Morristown, NJ, USA, 2003.

[2] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 1998.

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

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

[5] Robert Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

[6] Ruy Milidiú and Julio Duarte. Boosting at start. In *International Conference on Artificial Intelligence and Applications*, Innsbruck, Austria, 2009. IASTED.

[7] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[8] Bernd Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, Cambridge MA, 1995.

[9] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*, pages 41–48, 1999.

[10] Ian H. Witten and Eibe Frank. Data mining: practical machine learning tools and techniques with java implementations. *SIGMOD Rec.*, 31(1):76–77, 2002.

[11] Ross J. Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning).* Morgan Kaufmann, January 1993.

[12] Kristin P. Bennett, Ayhan Demiriz, and Richard Maclin. Exploiting unlabeled data in ensemble methods. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–296, New York, NY, USA, 2002.