IMPROVING CLASSIFIER ACCURACY USING UNLABELED DATA

Thamar I. Solorio Olac Fuentes

Department of Computer Science
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro #1
Santa María Tonantzintla, Puebla, México

ABSTRACT

This paper describes an algorithm for improving classifier accuracy using unlabeled data. This is of practical significance given the high cost of obtaining labeled data, and the large pool of unlabeled data readily available. The algorithm consists of building a classifier using a very small set of previously labeled data, then classifying a larger set of unlabeled data using that classifier, and finally building a new classifier using a combined data set containing the original set of labeled data and the set of previously unlabeled data. The algorithm proposed here was implemented using three well known learning algorithms: feedforward neural networks trained with Backpropagation, the Naive Bayes Classifier and the C4.5 rule induction algorithm as base learning algorithms. Preliminary experimental results using 10 datasets from the UCI repository show that using unlabeled data improves the classification accuracy by 5% on average and that for 80% of the experiments the use of unlabeled data results in an improvement in the classifier's accuracy.

1. INTRODUCTION

One of the problems addressed by machine learning is that of data classification. Since the 1960's, many algorithms for data classification have been proposed. However, all learning algorithms suffer the same weakness: when the training set is small the classifier accuracy is low. Thus, these algorithms can become an impractical solution due to the need of a very large training set. In many domains, unlabeled data are readily available, but manual labeling is time-consuming, difficult or even impossible. For example, there are millions of text documents available on the world-wide web, but, for the vast majority, a label indicating their topic is not available. Another example is character recognition: gathering examples with handwritten characters is easy, but manual labeling each character is a tedious task. In astronomy, something similar occurs, thousands of spectra per night can be obtained with an automated telescope, but an astronomer needs several minutes to manually classify each spectrum.

Thus the question is: can we take advantage of the large pool of unlabeled data? It would be extremely useful if we could find an algorithm that allowed improving classification accuracy when the labeled data are insufficient. This is the problem addressed in this paper. We evaluated the impact of incorporating unlabeled data to the learning process using several learning algorithms. Experimental results show that the classifiers trained with labeled and unlabeled data are more accurate than the ones trained with labeled data only. This is the result of the overall averages from ten learning tasks.

Even though the interest in learning algorithms that use unlabeled data is recent, several methods have been proposed. Blum and Mitchell proposed a method for combining labeled and unlabeled data called co-training [1]. This method is targeted to a particular type of problem: classification where the examples can naturally be described using several different types of information. In other words, an instance can be classified using different subsets of the attributes describing that instance. Basically, the co-training algorithm is this: two weak classifiers are built, each one using different kind of information, then, bootstrap from these classifiers using unlabeled data. They focused on the problem of web-page classification where each example can be classified using the words contained in that page or using the links that point to that page.

Nigam et al. proposed a different approach, where a theoretical argument is presented showing that useful information about the target function can be extracted from unlabeled data [2]. The algorithm learns to classify text from labeled and unlabeled documents. The idea in Nigam's approach was to combine the Expectation Maximization algorithm (EM) with the Naive Bayes classifier. They report an error reduction of up to 30%. In this work we extended this approach, incorporating unlabeled data to three different learning algorithms, and evaluate it using several data sets form the UCI Repository [3].

Unlabeled data have also been used for improving the performance of artificial neural networks. Fardanesh and Okan used the backpropagation algorithm, and the results show that the classifier error can be decreased using unlabeled data in some problem domains [4].

The paper is organized as follows: the next section presents the learning algorithms. Section 3 describes how unlabeled data are incorporated to the classifier's training. Section 4 presents experimental results that compare the performance of the algorithms trained using labeled and unlabeled data to those obtained by the classifiers trained with labeled data only. Finally, some conclusions and directions for future work are presented.

2. LEARNING ALGORITHMS

Experiments in this work were made with three of the most successful classification learning algorithms: feedforward neural networks trained with backpropagation, the C4.5 learning algorithm [5] and the Naive Bayes classifier.

2.1 Backpropagation and Feedforward Neural Networks

For problems involving real-valued attributes, Artificial Neural Networks (ANNs) are among the most effective learning methods currently known. Algorithms such as Backpropagation use gradient descent or other optimization algorithm to tune network parameters to best fit a training set of input-output pairs. The Backpropagation algorithm was applied in this work to a feedforward network containing two layers of sigmoidal units.

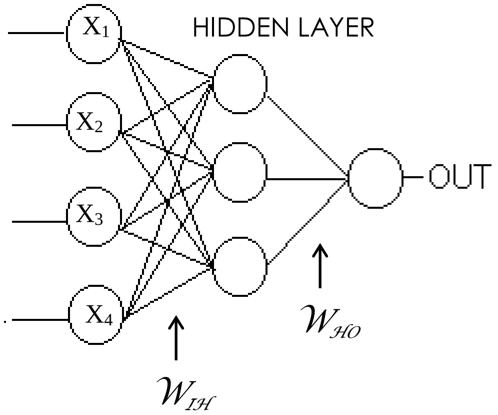

Figure 1. Representation of a feedforward neural network with one hidden layer.

2.2 Naive Bayes Classifier

The Naive Bayes classifier is a probabilistic algorithm based on the simplifying assumption that the attribute values are conditionally independent given the target values. Even though we know that in practice this assumption does not hold, the algorithm's performance has been shown to be comparable to that of neural networks in some domains [6,7]. The Naive Bayes classifier applies to learning tasks where each instance x can be described as a tuple of attribute values <a_1, a_2, … a_n> and the target function *f(x)* can take on any value from a finite set $\mathcal{V}$.

When a new instance x is presented, the Naive Bayes classifier assigns to it the most probable target value by applying this rule:

$$\mathcal{F}(x)=argmax_{v_j\in V}\mathcal{P}(v_i)\prod_I\mathcal{P}(a_i|v_j)$$

To summarize, the learning task of the Naive Bayes is to build a hypothesis by estimating the different $\mathcal{P}(v_i)$ and $\mathcal{P}(a_i|v_j)$ terms based on their frequencies over the training data.

2.3 The C4.5 Algorithm

C4.5 is an extension to the decision-tree learning algorithm ID3 [8]. Only a brief description of the method is given here, more information can be found in [5]. The algorithm consists of the following steps:

1. Build the decision tree form the training set (conventional ID3).
2. Convert the resulting tree into an equivalent set of rules. The number of rules is equivalent to the number of possible paths from the root to a leaf node.
3. Prune each rule by removing any preconditions that result in improving its accuracy, according to a validation set.
4. Sort the pruned rules in descending order according to their accuracy, and consider them in this sequence when classifying subsequent instances.

Since the learning tasks used to evaluate this work involve nominal and numeric values, we implemented the version of C4.5 that incorporates continuous values.

3. INCORPORATING UNLABELED DATA

The algorithm for combining labeled and unlabeled data is described in this section. In the three learning algorithms we apply this same procedure. First, the data set is divided randomly into several groups, one of these groups is considered with its original classifications as the training set, another group is separated as the test set and the remaining data are the unlabeled examples. A classifier C_1 is built using the training set and the learning algorithm L_1. Then, we use C_1 to classify the unlabeled examples. With the labels assigned by C_1 we merge both sets into one training set to build a final classifier C_2. Finally, the test data are classified using C_2.

The process described above was carried out ten times with each learning task and the overall averages are the results described in the next section.

4. EXPERIMENTAL RESULTS

We used the following dataset form the UCI repository: wine, glass, chess, breast cancer, lymphography, balloons, thyroid disease, tic-tac-toe, ionosphere and iris. Figure 2 compares the performance of C4.5 trained using the labeled data only with the same algorithm using both labeled and unlabeled data as described in the previous section. One point is plotted for each of the ten learning tasks taken from the Irving repository of machine learning datasets [2]. We can see that most points lie above the dotted line, which indicates that the error rate of the C4.5 classifier trained with labeled and unlabeled data is smaller than the error of C4.5 trained with labeled data only. Similarly, Figure 3 compares the performance of the Naive Bayes classifier trained using labeled and unlabeled data to that obtained using only labeled data. Again, a lower degree of error can be attained incorporating unlabeled data. Finally, Figure 4 shows the performance comparison of incorporating unlabeled data to a neural network's training to that using only labeled data.

As we can see in the three figures, the algorithm that shows the largest improvement with the incorporation of unlabeled data is C4.5. In the ten learning tasks C4.5 presented an improvement average of 8% while the improvement averages for neural networks and Naive Bayes were 5% and 3% respectively. Table 1 summarizes the results obtained in these experiments.

5. CONCLUSIONS AND FUTURE WORK

We have shown how learning from small sets of labeled training data can be improved upon with the use of larger sets of unlabeled data. Our experimental results using several training sets and three different learning algorithms show that for the vast majority of the cases, using unlabeled data improves the quality of the predictions made by the algorithms. This is of practical significance in domains where unlabeled data are readily available, but manual labeling may be time-consuming, difficult or impractical. Present and future work includes:

- Applying this methodology using ensembles of classifiers, where presumably the labeling of the unlabeled data and thus the final classifications assigned by the algorithm can be made more accurate.
- Experimental studies to characterize situations in which this approach is not applicable. It is clear that when the set of labeled examples is large enough or when the pseudo-labels can not be assigned accurately, the use of unlabeled data can not improve and may even decrease the overall classification accuracy.

	Naive Bayes Classifier			**Neural Networks**			**C4.5**		
	C1	C2	Ratio	C1	C2	Ratio	C1	C2	Ratio
Wine	11.11	**6.31**	**0.57**	7.61	7.57	0.99	22.34	20.56	0.92
Glass	69.35	68.82	0.99	**25.23**	26.21	1.04	61.04	58.60	**0.96**
Chess	28.35	37.91	1.34				19.58	**18.53**	**0.95**
Breast	27.30	27.51	1.01	5.94	**5.36**	**0.90**	10.70	10.28	0.96
Lympho	**27.82**	36.72	1.32				40.17	38.28	**0.95**
Balloons	28.43	32.18	1.13				32.50	**25.00**	**0.77**
Tiroides	9.31	8.44	0.91	10.00	**8.18**	**0.82**	18.97	18.46	0.97
tic_tac_toe	34.87	32.98	**0.95**				20.94	**19.81**	**0.95**
ionosphere	64.04	64.04	1.00	12.52	**12.47**	1.00	25.64	21.81	**0.85**
Iris	4.93	**2.64**	**0.54**	9.80	9.42	0.96	18.10	16.76	0.93
Average			0.97			0.95			0.92

Table 1. Comparison of the error rates of the three algorithms. C1 is the classifier built using labeled data only, C2 is the classifier built combining labeled and unlabeled data. Column Ratio presents results for C2 divided by the corresponding figure for C1. In bold we can see lowest error for a given dataset and the largest reduction in error as a fraction of the original error for each learning task. C4.5 shows the best improvement in 60% of the tasks. In 77% of the learning tasks the error was reduced when using unlabeled data, and in 80% of the tasks the best overall results where obtained by a classifier that used unlabeled data.

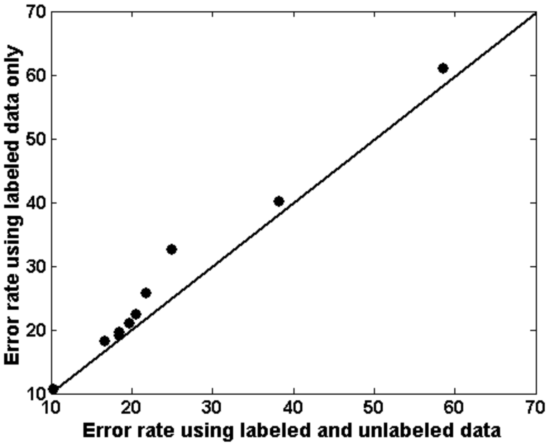

Figure 2. Comparison of C4.5 using labeled data only with C4.5 using unlabeled data. Points above the diagonal line exhibit lower error when the C4.5 is given unlabeled data.

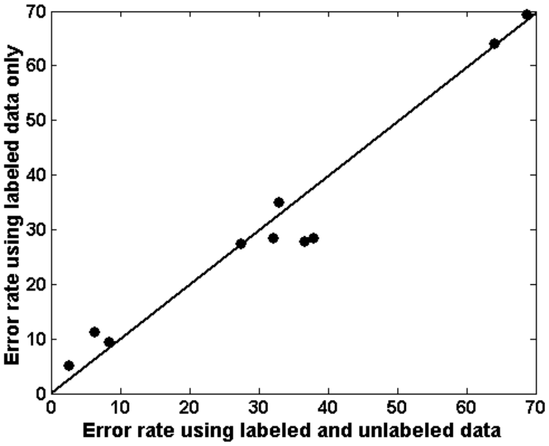

Figure 3. Comparison of Naive Bayes Classifier using labeled data only with Neural Network trained with a Naive Bayes Classifier using unlabeled data. Points above the diagonal line exhibit lower error when the Neural Network is given unlabeled data

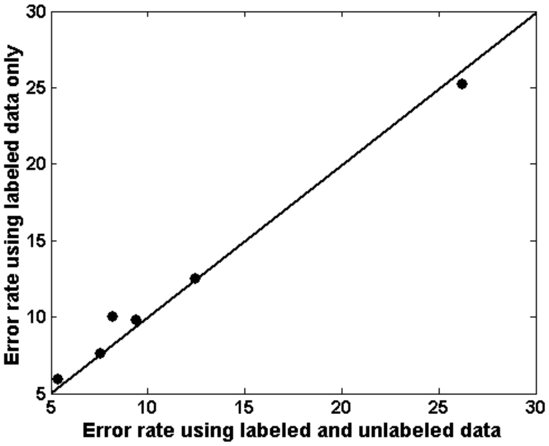

Figure 4.Comparison results of an ANN. Points above the diagonal line exhibit lower error when the ANN is given unlabeled data.

6. ACKNOWLEDGEMENT: We would like to thank CONACyT for partially supporting this work under grant J31877-A.

7. REFERENCES

[1] A. Blum, T. Mitchell, Combining Labeled and Unlabeled Data with Co-Training. *Proc. 1998 Conference on Computational Learning Theory*, July 1998.

[2] K. Nigam, A. McCallum, S. Thrun , & T. Mitchell, Learning to Classify Text from Labeled and Unlabeled Documents, *Machine Learning*, 1999,1-22.

[3] C. Merz, & P. M. Murphy, *UCI repository of machine learning databases*, http://www.ics.uci.edu./~mlearn/MLRepository.html, 1996.

[4] M.T. Fardanesh and Okan K. Ersoy, "Classification Accuracy Improvement of Neural Network Classifiers by Using Unlabeled Data," *IEEE Transactions on Geoscience and Remote Sensing* , Vol. 36, No. 3, 1998, 1020-1025.

[5] J. R. Quinlan, *C4.5: Programs for Machine Learning* (San Mateo, CA: Morgan Kaufmann, 1993).

[6] D. Lewis, & M. Ringuette, A comparison of two learning algorithms for text categorization, *Third Annual Symposium on Document Analysis and Information Retrieval*, 1994, 81-93.

[7] T. Joachims, A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, Proc. *1997 International Conference on Machine Learning*.1997.

[8] J. R. Quinlan, *Induction of decision trees.* Machine Learning, 1(1), 1986, 81-106.