Available online at www.sciencedirect.com
ScienceDirect
Knowledge-Based Systems 20 (2007) 220–224
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

Improving classification performance using unlabeled data: Naive Bayesian case

Chang-Hwan Lee *

Department of Information and Communications, DongGuk University, 3-26 Pil-Dong, Chung-Gu, Seoul 100-715, Republic of Korea

Received 11 July 2005; accepted 3 May 2006
Available online 8 September 2006

Abstract

In many applications, an enormous amount of unlabeled data is available with little cost. Therefore, it is natural to ask whether we can take advantage of these unlabeled data in classification learning. In this paper, we analyzed the role of unlabeled data in the context of naive Bayesian learning. Experimental results show that including unlabeled data as part of training data can significantly improve the performance of classification accuracy.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Machine learning; Semi-supervised learning; Naive Bayesian; Classification

1. Introduction

In current supervised learning methods, the algorithms need significant amount of labeled data to obtain satisfactory classification performance. Pure supervised learning from limited training data set will have poor classification performance [4].

However, in many domain applications, labeling is often done by a person, which is a time-consuming process. For example, in the task of classifying text documents, most of users of a practical system would not have the patience to label thousand articles. In fact, in many instances, the labeling process is harder or more expensive than the sampling step required to obtain the observations. Sometimes it is impossible to label most samples, and in some cases it is desirable to keep to a minimum the number of labeled samples.

Therefore, the need for large quantities of data to obtain high accuracy, and the difficulty of obtaining labeled data, raises an important question. What other sources of information can reduce the need for labeled data?

Most computational models of supervised learning rely on labeled training data, and ignore the possible role of unlabeled data. Therefore, situations in which both labeled and unlabeled samples are available arise naturally, and the investigation of the simultaneous use of both kinds of observations in learning leads immediately to questions of the relative value of labeled and unlabeled samples [7].

The purpose of this paper is to explore and study some techniques for improving the accuracy of classification algorithms by utilizing unlabeled data that may be available in large numbers and with no extra cost. We are to show how unlabeled data can improve the performance of classification learning in the context of naive Bayesian classifier. We present an algorithm and experimental results demonstrating that unlabeled data can significantly improve learning accuracy in certain learning problems. We identify the abstract problem structure that enables the algorithm to successfully utilize this unlabeled data, and show that unlabeled data will boost learning accuracy for problems in this class.

1.1. The value of unlabeled data

This section describes how unlabeled data are useful when learning classification. The unlabeled data are considered incomplete because they come without class labels.

* Tel.: +82 2 2260 3801; fax: +82 2 2285 3343.
E-mail address: chlee@dgu.ac.kr

0950-7051/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2006.05.014

How is it that unlabeled data can increase classification accuracy? At first consideration, one might be inclined to think that nothing is to be gained by access to unlabeled data. However, they provide information about the joint probability distribution.

Unlabeled data contain information about the joint distribution over features. If the probabilistic structure of data distribution is known, parameters of probabilistic models can be estimated by unsupervised learning, but it is still impossible to assign class labels without labeled data. This fact suggests that labeled data can be used to label the class and unlabeled data can be used to estimate the parameters of generative models.

It is known that unlabeled data alone are generally insufficient to yield better-than-random classification because there is no information about the class label [2]. However, unlabeled data do contain information about the joint distribution over features other than the class label. Because of this, they can sometimes be used, together with a sample of labeled data, to significantly increase classification accuracy in certain problem settings [14,8].

To see this, consider a simple classification problem, one in which instances are generated using a Gaussian mixture model [12]. Here, data are generated according to two Gaussian distributions, one per class, whose parameters are unknown. Fig. 1 illustrates the Bayes optimal decision boundary, which classifies instances into the two classes. Note that it is possible to calculate d from Bayes rule if we know the Gaussian mixture distribution parameters (i.e., the mean and variance of each Gaussian, and the mixing parameter between them). Consider when an infinite amount of unlabeled data is available, along with a finite number of labeled samples. It is well known that unlabeled data alone, when generated from a mixture of two Gaussians, are sufficient to recover the original mixture components [15]. However, it is impossible to assign class labels to each of the Gaussian without any labeled data. Thus, the remaining learning problem is the problem of assigning class labels to the two Gaussians.

For instance, in Fig. 1, the means, variances, and mixture parameter can be learned with unlabeled data alone. Labeled data must be used to determine which Gaussian belongs to which class. This problem is known to converge exponentially quickly in the number of labeled samples [2]. Informally, as long as there are enough labeled examples to determine the class of each component, the parameter estimation can be done with unlabeled data alone [2,15].

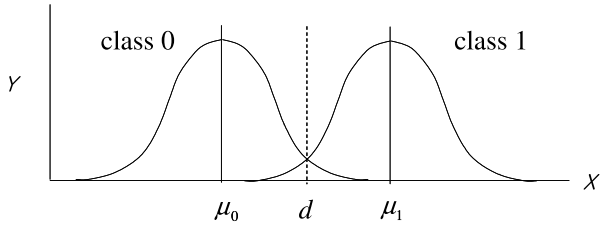

Fig. 1. Gaussian mixture model.

It is important to notice that this result depends on the critical assumption that the data indeed have been generated using the same parametric model as used in classification. Many applications require classifiers built by machine learning to categorize incoming data automatically. Since in many applications, an enormous amount of unlabeled data is available with little cost, it is natural to ask whether, in addition to human labeled data, one can also take advantage of the unlabeled data.

1.2. Related work

Most classic methods of learning with unlabeled data use a generative model for the classifier and use Expectation-Maximization (EM) method. The EM algorithm was first introduced in [5] as a procedure for estimating the parameter values that maximize the likelihood function when there are missing data. There are two steps in EM algorithm, expectation (E-step) and maximization (M-step). The E-step calculates the expected values of the sufficient statistics given the current parameter estimates. The M-step makes maximum likelihood estimates of the parameters given the estimated values of the sufficient statistics. Starting with a randomly generated set of parameter estimates, these steps alternate until the parameter estimates in iteration $k-1$ and k differ by less than ϵ or until a predefined maximum number of iterations has been exceeded.

In co-training method [11,1,13], two classifiers are defined using two distinct feature sets. Co-training then consists of iteratively using the output of one classifier to train the other. Variations of co-training have been applied to many application areas. Co-training is a learning algorithm in which the redundancy of the learning task is captured by training two classifiers using separate views of the same data. This enables bootstrapping from a small set of labeled training data via a large set of unlabeled data. In co-training, two classifiers are defined using distinct feature sets. Co-training then consists of iteratively using the output of one classifier to train the other. Variations of co-training have been applied to many applications areas.

Another related area of research that has very different goal is that of active learning [3,9] where the algorithm repeatedly selects an unlabeled example, asks an expert to provide the correct label, and then rebuilds its hypothesis. An important component of active learning is in the selection of the unlabeled example.

2. Problem formulation

This section presents a probabilistic framework for characterizing the nature of classifiers. This paper employs naive Bayes – a well-known probabilistic classifier [6,10], as the foundation upon which we will later build in order to incorporate unlabeled data.

The naive Bayes method applies to learning tasks where each instance x is described by a conjunction of attribute values and where the target function can take on any value

from some finite set V. A set of training examples of the target function is provided, and a new instance is presented, described by the tuple of attribute values $(a_1, a_2, \ldots, a_n)$. The learner is asked to predict the target value, or classification, for this new instance.

The Bayesian approach to classifying the new instance is to assign the most probable target value, given the attributes $(a_1, a_2, \ldots, a_n)$ that describe the instance. Suppose a_i denote the ith attribute. The naive Bayes expression for the probability of a new instance given its class is given as

$$v = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j).$$

Thus the parameters of an individual mixture components are a multinomial distribution. The only other parameters of the model are mixture weights(class prior probabilities), which indicate the probabilities of selecting the different mixture components. Thus the complete collection of model parameters is a set of multinomials and prior probabilities over those multinomials.

Learning a naive Bayes consists of estimating these parameters of the generative model by using a set of labeled training data.

Suppose the estimate of $P(a_i|v_j)$ and $P(v_j)$ are denoted as $\hat{p}_{ij}$ and α_j, respectively, then $\hat{p}_{ij}$ is defined as

$$\hat{p_{ij}} = \frac{P(a_i \wedge v_j)}{P(v_j)} = \frac{N(a_i \wedge v_j)}{N(v_j)}, \quad \hat{\alpha_j} = \frac{N(v_j)}{N}, \tag{1}$$

where $N(t)$ means the number of data satisfying the condition t.

The parameter estimation formulae in Eq. (1) that result from this maximization are the familiar ratios of empirical counts. Furthermore, the counts in both the numerator and denominator are augmented with pseudo-counts. The class prior probabilities are estimated in the same manner, and also involve a ratio of counts with smoothing. The use of this type of prior is sometimes referred to as Laplace smoothing. Smoothing is necessary to prevent zero probabilities for infrequently occurring. Therefore, in this paper, we use Laplace smoothing and, thus, the final probability estimates are given as:

$$\hat{p_{ij}} = \frac{N(a_i \wedge v_j)}{N(v_j) + N}, \quad \hat{\alpha_j} = \frac{N(v_j)}{N + C}, \tag{2}$$

where N represents the total number of data and C represents the number of class values, respectively.

Given estimates of these parameters calculated from the training data according to Eq. (2), it is possible to turn the generative model backwards and calculate the probability that a particular mixture component generated a given data.

2.1. Incorporating unlabeled data

We have modified the naive Bayesian algorithm in order to process the unlabeled data. The basic mechanism of processing unlabeled data is as follows. The algorithm first calculates the estimates of parameters using only labeled data. After acquiring estimated values of parameters, the algorithm classifies unlabeled data. It calculates the weights of each value of the target feature, and the target value with the highest weight is assigned to the value of the target attribute. After the class values of each unlabeled data are calculated, the algorithm is trained again using both originally labeled data and formerly unlabeled data. The algorithm iterates this process until there is no or very little changes in the estimated target values of the unlabeled data. Fig. 2 represents the brief pseudo-code of the algorithm.

```
Given : L : labeled data, U : unlabeled data

/* process labeled data */
calculate P̂_ij and α̂_j and using all data in L
while (no change in the values of P̂_ij and α̂_j) do
    /* process unlabeled data */
    for each data (a_1, ..., a_t) ∈ U do
        calculate v satisfying v = max_j α̂_j ∏_i P̂_ij
        assign label v to (a_1, ..., a_t)
    end-for
    calculate P̂_ij and α̂_j using both L and U
end-do
```

Fig. 2. Pseudo-code of the algorithm for incorporating unlabeled data.

3. Empirical studies

In this section, we present experimental results of the influence of unlabeled data using the proposed algorithm. We have tested the effectiveness of unlabeled data using a set of synthetic data to precisely analyze the effect of estimating parameters.

The data are generated according to multiplicative distribution. For each attribute i, two parameters, p_i and q_i are associated with the attribute. The parameter p_i represents the probability that attribute i becomes 1 given the class value is 0. Similarly, q_i represents the probability that attribute i becomes 1 given the class value is 1. There is also another parameter, α, which represents the probability that class 0 is selected.

$$\alpha = P(v = 0), \quad p_i = P(a_i = 1|v = 0), \quad q_i = P(a_i = 1|v = 1).$$

The values of these parameters are determined as given in Fig. 3. Using these values, we generated a set of labeled data as well as unlabeled data.

Fig. 4 shows the pseudo-code of the algorithm for generating data set. We could also calculate the accuracy of optimal classifier in case test data are generated based on the algorithm in Fig. 5.

$\alpha = 0.5$
$p_1 = 0.7$, $p_2 = 0.3$, $p_3 = 0.8$, $p_4 = 0.3$, $p_5 = 0.2$, $p_6 = 0.1$, $p_7 = 0.8$
$q_1 = 0.2$, $q_2 = 0.6$, $q_3 = 0.3$, $q_4 = 0.3$, $q_5 = 0.6$, $q_6 = 0.7$, $q_7 = 0.8$

Fig. 3. Parameter values for generating sample data.

Given : α : prob. that class 0 is selected
p_i : prob. that i-th attribute = 1 when target value =0
q_i : prob. that i-th attribute = 1 when target value =1
t : number of attributes, n : number of data, D : the data set

$D := \{\}$
repeat n **times**
  determine the class value v ($v = 0$ with probability α)
  for j=1 **to** t
    if v=0 **then**
      determine the value of α_j ($\alpha_j = 1$ with probability p_i)
  else if v=1 **then**
      determine the value of α_j ($\alpha_j = 1$ with probability q_i)
    end-if
  $D := D \cup (a_1, \ldots, a_t, v)$;
end-repeat
return D

Fig. 4. Algorithm for generating sample data.

Given : α : prob. that class 0 is selected
p_i : prob. that i-th attribute = 1 when target value =0
q_i : prob. that i-th attribute = 1 when target value =1
t : number of attributes, D : the data set

correct := incorrect := 0
for each tuple $(a_1, \ldots, a_t, v) \in D$ **do**
  $\text{Sum}_0 := \alpha$; $\text{Sum}_1 := 1 - \alpha$;
  for i=1 **to** t
    if v=0 **then**
      if $\alpha_i = 1$ **then** $\Delta_i = p_i$ **else** $\Delta_i = 1 - p_i$ **end-if**
      $\text{Sum}_0 := \text{Sum}_0 * \Delta_i$;
    else if v=1 **then**
    if v=0 **then**
      if $\alpha_i = 1$ **then** $\delta_i = q_i$ **else** $\delta_i = 1 - q_i$ **end-if**
      $\text{Sum}_1 := \text{Sum}_1 * \delta_i$;
    end-if
  end-for
  if $\text{Sum}_0 > \text{Sum}_1$ **then** output := 0 **else** output := 1 **end-if**
  if output = v **then** correct++ **else** incorrect++ **end-if**
end-for
return correct/(correct+incorrect);

Fig. 5. Algorithm for calculating optimal classification accuracy.

The size of test data is given as 1000, disjoint with training data, and the optimal Bayes accuracy using the values in Fig. 3 is calculated as 86.6%.

The proposed algorithm first trains a classifier with only the available labeled data, and uses the classifier to assign labels of each unlabeled data by calculating the expectation of the missing class labels. It then trains a new classifier using all the data – both the originally labeled and the formerly unlabeled – and iterates. Over the course of several experimental comparisons, we show that unlabeled data can significantly increase performance.

Fig. 6 shows the effect of using unlabeled data in case the number of labeled data is given as 10, 20 and 30, respectively. As shown in Fig. 6, when the number of labeled data are small, the classification accuracy could be greatly improved by incorporating unlabeled data. We vary the amount of unlabeled data from 0 to 100,000 and see how the effect

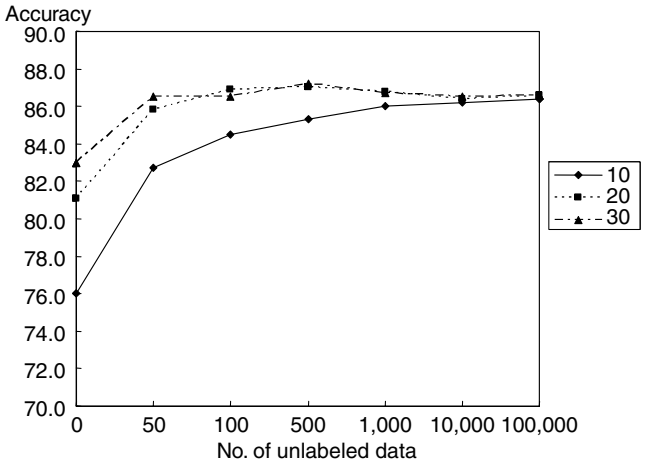

Fig. 6. Accuracies based on the number of unlabeled data (1).

of using unlabeled data. When only 10 labeled data are given, the classification accuracy is initially given as 76%. However, as the amount of unlabeled data increases, the accuracy increases as well. The accuracy reaches 86.6%, the optimal accuracy for given data set, when sufficient amount (100,000 in this experiment) of unlabeled data is given.

Fig. 7 shows the effect of using unlabeled data in case the number of labeled data is moderate (e.g., 50, 100 and 500, respectively). The classification accuracy increases as more unlabeled data are available. However, in Fig. 7, we see that initial accuracies using labeled data are higher than that of Fig. 6 since larger number of labeled data are available in Fig. 7. Furthermore, the system required less number of unlabeled data in order to approach the optimal accuracy (86.6%), and the effect of unlabeled data in the case of Fig. 7 is less significant than that of Fig. 6.

Fig. 8 shows the effect of using unlabeled data in case the number of labeled data is large (e.g., 1000, 5000 and 10,000, respectively). In these cases, since the system already have enough number of labeled data, the use of

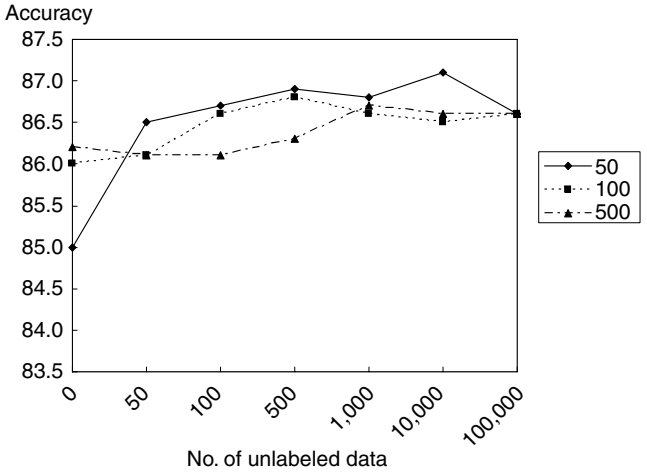

Fig. 7. Accuracies based on the number of unlabeled data (2).

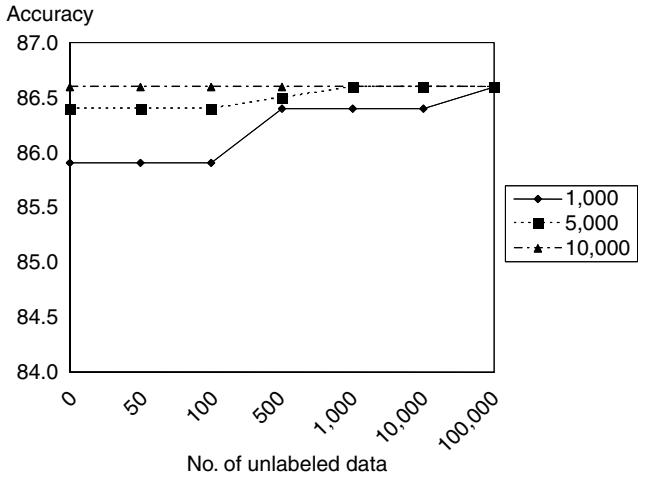

Fig. 8. Accuracies based on the number of unlabeled data (3).

unlabeled data does not show any significant improvement in terms of classification accuracy. When 10,000 cases of labeled data are given, the system already reached the optimal accuracy.

As we can see in Fig. 6, the reduction in the number of labeled data needed can be significant. In order to identify the data with the optimal accuracy (86.6%), a traditional learner, with no unlabeled data, requires more than 5000 labeled data; alternatively our algorithm takes advantage of unlabeled data and requires only 10 labeled data along with 10,000 unlabeled data to achieve the same accuracy. Thus, in this task, the proposed method in this paper significantly reduces the need for labeled training data. With only 10,000 labeled data, accuracy is improved from 76.0% to 86.6% by adding unlabeled data. These findings illustrate the power of unlabeled data, and also demonstrate the strength of the algorithms proposed here.

4. Conclusion

In this paper, we demonstrate the effectiveness of using unlabeled data in classification learning in the context of naive Bayesian method. We proposed a semi-naive Bayesian learning method, which could incorporate unlabeled data in naive Bayesian learning. The algorithm was tested on a set of synthesized data.

The experimental results show that the proposed algorithm could effectively take advantage of unlabeled data in classification learning. Compared with pure supervised learning method, the algorithm needs significantly smaller amount of labeled data, along with large number of unlabeled data, to achieve the optimal classification accuracy.

References

[1] Avrim Blum, Tom Mitchell, Combining Labeled and Unlabeled Data with Co-Training, in: COLT, 1998.
[2] Vittorio Castelli, Thomas M. Cover, On the exponential value of labeled samples, Pattern Recognition Letters 16 (1995) 105–111.
[3] D. Cohn et al., Active learning with statistical models, Journal of Artificial Intelligence Research 4 (1996) 129–145.
[4] F. De Comite et al., Positive and unlabeled examples help learning, in: Tenth International Conference on Algorithmic Learning Theory, 1999, pp. 219–230.
[5] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of Royal Statistical Society 39 (1977) 1–38.
[6] R. Duda et al., Pattern Classification, second ed., Wiley, New York, 2001.
[7] Sally Goldman, Yan Zhou, Enhancing supervised learning with unlabeled data, in: International Conference on Machine Leaning, 2000.
[8] T. Hofmann, Text categorization with labeled and unlabeled data: a generative model approach, in: NIPS 99 Workshop on Using Unlabeled Data for Supervised Learning, 1999.
[9] R. Liere, P. Tadepalli, Active learning with committees for text categorization, in: 14th National Conference on Artificial Intelligence, 1997, pp. 591–596.
[10] Tom Mitchell, Machine Learning, McGraw Hill, New York, 1997.
[11] T. Mitchell, The role of unlabeled data in supervised learning, in: 6th International Colloquium on Cognitive Science, 1999.
[12] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Machine Learning 39 (2000) 103–134.
[13] Kamal Nigam, Rayid Ghani, Analyzing the effectiveness and applicability of co-training, in: CIKM 2000.
[14] B. Shahshahani, D. Landgrebe, The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon, IEEE Transactions on Geoscience and Remote Sensing 2 (5) (1994) 1087–1095.
[15] T. Zhang, Some asymptotic results concerning the value of unlabeled data, in: NIPS 99 Workshop on Using Unlabeled Data for Supervised Learning, 1999.