B-EM: A Classifier Incorporating Bootstrap with EM Approach for Data Mining

Xintao Wu
UNC at Charlotte
9201 Univ. City Blvd
Charlotte, NC 28223
xwu@uncc.edu

Jianping Fan
UNC at Charlotte
9201 Univ. City Blvd
Charlotte, NC 28223
jfan@uncc.edu

Kalpathi R. Subramanian
UNC at Charlotte
9201 Univ. City Blvd
Charlotte, NC 28223
krs@uncc.edu

ABSTRACT

This paper investigates the problem of augmenting labeled data with unlabeled data to improve classification accuracy. This is significant for many applications such as image classification where obtaining classification labels is expensive, while large unlabeled examples are easily available. We investigate an Expectation Maximization (EM) algorithm for learning from labeled and unlabeled data. The reason why unlabeled data boosts learning accuracy is because it provides the information about the joint probability distribution. A theoretical argument shows that the more unlabeled examples are combined in learning, the more accurate the result. We then introduce B-EM algorithm, based on the combination of EM with bootstrap method, to exploit the large unlabeled data while avoiding prohibitive I/O cost. Experimental results over both synthetic and real data sets show that the proposed approach has a satisfactory performance.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous

Keywords

Expectation Maximization, Classification, Supervised and Unsupervised learning, Bootstrap Method

1. INTRODUCTION

Classification has been identified as an important problem in data mining field. There has been focus on algorithms [17, 8, 13] that can build classifier over large labeled training sets. The intuition there is that by building classifier over large training data sets, we will be able to improve the accuracy of the classification model. One key difficulty with these current algorithms is that the assumption requires a large, often prohibitive, number of labeled training records to learn accurately.

However, in many modern classification applications such as text categorization, web categorization and image classification, the training datset may not be assumed to be fully labeled. On the contrary, the training data set usually may contain only very few labeled examples and a large number of unlabeled examples due to the fact hand-labeling is expensive while collecting unlabeled records is trivial especially for those involving online sources.

Consider the problem of training an image classifier to automatically classify the images. Given the growing volume of online images available through the internet, this problem is of great practical significance. Each image is associated with a large number of visual features such as color, texture, shape etc which can be extracted automatically by feature extraction tools. Classification problem here involves learning a mapping from a known feature space to a set of discrete semantic class labels.

In this paper, we investigate an algorithm that learns to classify data more accurately by combining unlabeled records to augment the available labeled training records. The reason why unlabeled samples boost learning accuracy is, in brief, the unlabeled samples provide the information about the joint probability distribution over feature values of the records [4]. Here, we assume the labeled training examples may only be a small fraction of total training data set. The labeled part can be easily fitted in memory while the total volume of training set may be too large to be fitted in memory. We argue those traditional classification algorithms such as neural networks [12], statistical models [11], decision trees [16] and genetic models [10] suffer the accuracy here as they assume a sufficient set of labeled training data. The specific approach we describe here is to extend conventional learning algorithms over large data sets by using Expectation Maximization (EM) to dynamically derive pseudo-labels for unlabeled image into supervised learning. For cases where a large volume of unlabeled records is available, we propose a novel algorithm, B-EM, based on a combination of two well known learning algorithms: the bootstrap method [5] and EM [7] algorithm. B-EM greatly reduces the number of database scans while achieving the satisfactory accuracy (bounded in a small range with a high confidence level).

The remainder of the paper is structured as follows. In Section 2, we survey existing work on EM in supervised and

SIGKDD '02 Edmonton, Alberta, Canada
Copyright 2002 ACM 1-58113-567-X/02/0007 ...$5.00.

unsupervised learning. In Section 3, we formally introduce how to extend EM over labeled and unlabeled training data. We present our new algorithm B-EM which is applicable to large training data set. In Section 4, we present experimental results and performance evaluation over both synthetic and real data sets. We present the conclusion and address the future work in Section 5.

2. RELATED WORK

The EM algorithm [7] is a general technique for finding maximum likelihood estimates for parametric models when the data are not fully observed. EM has been well studied for unsupervised learning and has been shown to be superior to other alternatives for statistical modeling purposes [6]. The well-known AutoClass project[6] investigates the combination of the EM algorithm with naive bayes classifier where they emphasize how to discover clusters for unsupervised learning over unlabeled data. Bradley et al. present a scalable clustering framework where they apply EM algorithm to the data summary instead of the original data [2]. The algorithm presented effectively scales to very large databases as it requires at most one scan of the database. However, it is unknown how to apply the above algorithm for supervised learning.

Ghahramani et al. present a framework [9] based on maximum likelihood density estimation where EM is applicable both for supervised and unsupervised learning problems. For example, by estimating the joint density of the input and class label using a mixture model, the classification problems can be thought learning a mapping from an input space into a set of discrete class labels.

The theoretical work [3, 4] shows use of unlabeled data can improve parameter estimates of mixture model. The results can be highlighted as following 1) unlabeled data does not improve the classifiaton results in the absence of labeled data; 2) the classification error approaches the bayes optimal solution at an exponential rate in the number of labeled examples given if infinite amounts of unlabeled data are available; 3) the labeled data can be exponentially more valuable than unlabeled data in reducing the probability of classification error; 4) the additional unlabeled samples should always improve the performance.

Nigam et al. introduce an algorithm for learning from labeled and unlabeled text, based on the combinations of EM with a naive bayes classifer and show that the accuracy of learned text classifiers be improved by augumenting a small number of labeled training documents with a large pool of unlabeled documents [14].

3. OUR METHOD

3.1 General Approach and Notation

We are given n records in a training set $S = S^l \cup S^u$. Each record s_i takes the form $s_i =< \mathbf{x}_i, y_i >$, where $\mathbf{x}_i$ is an associated d-attribute vector, $< x_i^1, \cdots, x_i^d >$, which is depicting d measurements made on the data from d attributes, repsectively, $A^1, \cdots, A^{d1}$, y_i denotes the class label of records from m classes $C = \{c_1, \cdots, c_m\}$. The record $s_i \in S^l$ comes with the known class label $y_i \in C$, and for the rest of the records, in subset S^u, the class label y_i is unknown.

Now the learning task is, given $S = S^l \cup S^u$, how to build a classifer which can predict y_i based on $\mathbf{x}_i$ for new data $s_i \in S^t$, where S^t is test data sets . Note the traditional classifcation approach is to build the classifer only based on labeled training data S^l.

In this paper, we will apply a mixture model for characterizing the nature of the data and classifiers. The mixture model follows two commonly used assumptions about the data: 1) the data are produced by a mixture model; 2) each record only belongs to one class and there is a one-to-one correspondence between the components in the mixture model and classes.

$$P(\mathbf{x}_i|\theta) = \sum_{j=1}^{m} P(c_j)P(\mathbf{x}_i|c_j, \theta_j) \qquad (1)$$

The mixture model, as shown in Equation 1 has two parts: the first part gives the interclass mixture probability $P(c_j)$ that an example s_i is a member of class c_j, independently of anything else we may know of the data; the second part $P(\mathbf{x}_i|c_j, \theta_j)$, shows the data in each class c_j are then modeled by a class distribution (component), giving the probability of observing the instance attribute values $\mathbf{x}_i$, conditional on the assumption that instance s_i belongs in class c_j.

The interclass pdf is a Bernoulli distribution characterized by the class number m and the probabilities of each class. As the distribution of each class is unknown, the multidimensional Gaussian distribution is usually assumed for it provides a good approximation for unknown distributions. In this paper, we assume all attributes $\mathbf{x}_i$ are continuous variables[2].

Equation 2 shows a d-dimensional Gaussian distribution for class c_j, where $j = 1, \cdots, m$, μ_j is d-dimensional mean vector and Σ_j is $d \times d$ covariance matrix, the superscript T indicates transpose , $|\Sigma_j|$ is the determinant of Σ_j and Σ_j^{-1} is its matrix inverse.

$$P(\mathbf{x}|c_j) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_j|}} exp\{-\frac{1}{2}(\mathbf{x}-\mu_j)^T\Sigma_j^{-1}(\mathbf{x}-\mu_j)\} \qquad (2)$$

In this setting, each record, s_i, is created by first selecting a component according to priors $P(c_j)$, then, second, having the mixture component generate the record according to its own distribution $P(\mathbf{x}_i|c_j, \theta_j)$ with parameters θ_j[3].

Under this framework, classification problems can be solved by estimating joint density of the known attributes and class label using a mixture model and computing a maximum likelihood estimate of θ, i.e., finding the parameterization that is most likely given our S. EM is a widely used iterative technique which can concurrently generate probabilistically-assigned labels for the unlabeled data, and a more probable model with smaller parameter variance that predicts these same probabilistic labels.

3.2 EM Expansion with Unlabeled Examples

When traditional classifier approach such as naive bayesian, decision trees etc. is given a small set of labeled training

[1] x_i^j contains the relevant information required for measuring the similarity between the data.

[2] Discrete or categorical data can be modeled as generated by a mixture of multinomial densities and similar derivations for the learning algorithm can be applied.

[3] Under the assumption of Gaussian distribution, θ_j includes μ_j, Σ_j.

data, classification accuracy suffers. This section shows, by augmenting this small set with a large set of unlabeled data and combining the two sets with EM, we can improve the parameter estimates and hence classification accuracy.

Consider the probability of all the labeled and unlabeled training data, $S = S^l \cup S^u$ under the two part model framework. The probability of the whole data is simply the product over all the data shown as,

$$P(S|\theta) = \prod_{i=1}^{n} P(s_i|\theta)$$

here we assume one record is independent with the others.

The likelihood of an unlabeled record can be characterized as the sum of total probability over all mixture components.

$$P(\mathbf{x}|\theta) = \sum_{j=1}^{m} P(\mathbf{x}|c_j, \theta_j)P(c_j)$$

For the labeled record, we are given the label y_i and thus do not need to sum over all class components as shown,

$$P(\mathbf{x}|\theta) = P(\mathbf{x}|c_j, \theta_j)P(c_j)$$

When combining both labeled and unlabeled records, the probability of the whole training data set is shown as Equation 3.

$$\begin{aligned} P(S|\theta) &= \prod_{i=1}^{|S^u|} \sum_{j=1}^{m} P(c_j)P(\mathbf{x_i}|c_j, \theta_j) \\ &\times \prod_{i=1}^{|S^l|} P(c_j = y_i)P(\mathbf{x_i}|c_j, \theta_j). \end{aligned} \quad (3)$$

By the maximum likelihood principle, the best model of the data has parameters that maximize $P(\theta|S)$. Equation 4 shows the log likelihood of the parameters given the data set.

$$\begin{aligned} log(P(\theta|S)) &= log(P(\theta)/P(S)) + \\ & \sum_{i=1}^{|S^u|} log \sum_{j=1}^{m} P(c_j)P(\mathbf{x}_i|c_j, \theta_j) + \\ & \sum_{i=1}^{|S^l|} log(P(c_j = y_i)P(\mathbf{x}_i|c_j, \theta_j)) \end{aligned} \quad (4)$$

Note the first part is a constant for $P(S)$ is a constant and maximum likelihood estimation assumes that $P(\theta)$ is a constant.

However, the second part of this equation has a log of sums, it is not easily maximized numerically. Intuitively, it is unclear which component of the mixture model generated a given record and thus which parameters to adjust to fit the feature value of that record. When all the class labels in S^l are given, we could express this complete log likelihood of the parameters without a log of sums [9]. We introduce the binary indicator z_{ij} where $z_{ij} = 1$ iif $y_i = c_j$ else $z_{ij} = 0$ in Equation 5. By introducing a hidden variable $\mathbf{z}$ that indicate which record was generated by which component, then the maximization problem decouples into a set of simple maximizations.

$$\begin{aligned} log(P(\theta|S, \mathbf{z})) &= log(P(\theta)/P(S)) + \\ & \sum_{i=1}^{S} \sum_{j=1}^{m} z_{ij} log(P(c_j|\theta))P(\mathbf{x}_i|c_j, \theta_j) \end{aligned} (5)$$

Since $\mathbf{z}$ is unknown, the $log(P(\theta|S, \mathbf{z}))$ can not be utilized directly. The EM algorithm can be used to find a local maximum likelihood parameter by an iterative procedure through the following two steps.

- E-step, which corresponds to calculating probabilistic labels $P(c_j|\mathbf{x}_i, \theta)$ for every record by using the current estimate of $\hat{\theta}$. Equation 6 shows how to compute $E[z_{ij}|\mathbf{x}_i, \theta^{(k)}]$ (we denote as $h_{ij}^{(k)}$), the probability that Gaussian j, as defined by the parameters estimated at step k, generated data $\mathbf{x}_i$.
- M-step, which corresponds to calculating a new maximum likelihood estimate for parameter θ given the current estimates for $P(c_j|\mathbf{x}_i, \theta)$. As shown in Equation 7 and 8, the M-step re-estimates the means and covariances of the Gaussians using the data set weighted by the $h_{ij}^{(k)}$.

$$h_{ij}^{(k)} = \frac{|\hat{\Sigma}_j^k|^{-\frac{1}{2}} exp\{-\frac{1}{2}(\mathbf{x}_i - \hat{\mu}_j^{(k)})^T \hat{\Sigma}_j^{-1,(k)} (\mathbf{x}_i - \hat{\mu}_j^{(k)})\}}{\sum_{l=1}^{m} |\hat{\Sigma}_l^k|^{-\frac{1}{2}} exp\{-\frac{1}{2}(\mathbf{x}_i - \hat{\mu}_l^{(k)})^T \hat{\Sigma}_l^{-1,(k)} (\mathbf{x}_i - \hat{\mu}_l^{(k)})\}} \quad (6)$$

$$\hat{\mu}_j^{(k+1)} = \frac{\sum_{i=1}^{n} h_{ij}^{(k)} \mathbf{x}_i}{\sum_{i=1}^{n} h_{ij}^{(k)}} \quad (7)$$

$$\hat{\Sigma}_j^{(k+1)} = \frac{\sum_{i=1}^{n} h_{ij}^{(k)} (\mathbf{x}_i - \hat{\mu}_j^{(k+1)})(\mathbf{x}_i - \hat{\mu}_j^{(k+1)})^T}{\sum_{i=1}^{n} h_{ij}^{(k)}} \quad (8)$$

The parameter $\hat{\theta}$ generated by EM that locally maximizes the probability of all the data (both the labeled and unlabeled) will be used to label the test data with the largest posterior probability.

3.3 EM Over Large Unlabeled Data Sets

In this paper, we assume that the size of labeled data set S^l is small, hence, it can be easily fitted in memory. While the size of unlabeled data S^u can be too large to be fitted in memory, computing a mixture model over large databases via standard EM would not be acceptable as hundreds of iterations or more may be required during iterative EM refinement step. One straightforward approach is to sample unlabeled records as many as the memory can hold. As shown from theoretical work [3, 4], the more additional unlabeled data, the more accuracy we achieve. Although guaranteed to converge, a general bound on the number of unlabeled data required for a given training data set is not available. As shown in experiment results, the training data set generated by class distributions with larger variances needs more unlabeled data to converge. In this paper, we combine the EM with bootstrap methods [5] to classify with large sizes of training set.

The resulting B-EM algorithm is very straightforward and can be outlined at a high level as follows:

```
1. Build an initial classifier by estimating the parameters of model from the labeled data only.
2. Repeat $M$ times
    • Obtain a radom bootstrap sample from unlabeled data, filling in the memory buffer.
    • Repeat until the parameters $\theta^l$ do not change
        – Apply the current classifer to probabilistically label the unlabeled data in the buffer.
        – Recalculate the classifier parameters $\theta^l$ given the probabilistically assigned labels.
3. compute $\theta^* = \frac{1}{M}\sum_{l=1}^{M} \theta^l$.
4. Apply the $\theta^*$ to probabilistically label all the unlabeled data.
```

Ideally, we would like to say that θ^* is very close to the $\hat{\theta}$ computed over all records in training set. The bootstrap principle [5] shows the two estimates converges the same when bootstrap steps M is sufficiently large (e.g., 100).

We can see the B-EM needs at most two scans of data while achieving almost the same accuracy. When the training data set is totally unordered , we get bootstrap samples by randomly fetching from disk block. In this case, we only need one scan (the step 4 which labels all unlabeled data). When the training data set is not totally unordered, we need one more read scan to generate M bootstrap samples and write each samples to disk. In this case, the total number of scans is bounded by two. It is also worth pointing out B-EM also saves CPU cost. The explaination has two parts. First, for B-EM, the number of unlabeled data invloved in EM iterative steps is less than standard EM. Second, EM empirically tends to need fewer iterations with less data.

3.4 Discussion

In this paper, we also combine EM with naive bayesian approach which assumes the values of the attributes are conditionaly independent of one another when the number of dimensions is too large. The explaination has two parts. First, in estimating $P(c_j|\mathbf{x}_i, \theta)$ as shown in Equation 6, we need compute, Σ_l^{-1}, the inverse of covariance matrix. When the values of the attributes are not conditionaly independent, the covariance matrix Σ_l is singular which causes the estimation overflow. Second, we can reduce computation cost in evaluating $P(\mathbf{x}|c_j, \theta_j)$ involved in EM.

Under the naive bayesian assumption, the class distribution shows as,

$$P(\mathbf{x}|c_j, \theta_j) = \prod_{k=1}^{d} P(x^k|c_j, \theta_j)$$

The probabilities $P(x^k|c_j)$ can be estimated from the training sample, where each attribute A^k is assumed to have a normal distribution with mean $\mu_{c_j}^k$ and standard variance $\sigma_{c_j}^k$ respectively, given the values for attributes A^k for training samples of class c_j.

4. EXPERIMENTAL RESULTS

The experiments are conducted on a Dell PowerEdge 4400, with two processors and 1G bytes of RAM.

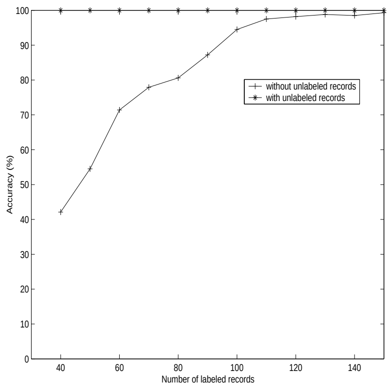

Figure 1: Classification accuracy on the synthetic data set DS1 (10k records, 10 classes), both with and without unlabeled images

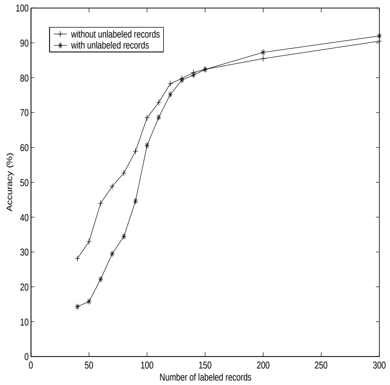

Figure 2: Classification accuracy on the synthetic data set DS4 (10k records, 10 classes), both with and without unlabeled images

4.1 Synthetic Data

4.1.1 Classification Quality Evaluation

In this section, we show using unlabeled training data in EM do improve the classification accuracy.

The synthetic data sets (DS1, DS2, DS3, DS4) we generate in this experiment consists of $m = 10$ classes. Each class is defined by a multi-dimensional Gaussian distribution ($d = 25$) which is characterized by mean vector μ and covariance matrix $\mathbf{\Sigma}$. For all these four data sets, the Gaussian means are chosen uniformly on $[0.0, 10.0]$. For DS1, the diagonal covariance matrics are chosen uniformly on $[0.8, 1.2]$. The distribution of DS1 is the same as the data set generated in [2] and it is expected the clusters are fairly separated. We then generate DS2, DS3 and DS4 by varying the variance (multiplying a constant 5, 8, 10). Note the larger the variance of multi-dimensional Gaussian distribution, the less separated as more outliers are generated by the Gaussian distributions with large variance.

For each data set, we generate 10k records (each class has 1k records). Varying sizes of random subsets are labeled and the remaining subsets are used as unlabeled records.

Figure 1 and 2 show the effect of using EM with unlabeled data over synthetic data sets DS1 and DS4 respectively. The vertical axis indicates the accuracy rate which is computed on the basis of the remaining unlabeled records, and the horizontal axis indicates the amount of labeled data used in training. We vary the amount of labeled training data, and compare the classification accuracy of EM without unlabeled data with EM with unlabeled data. EM with unlabeled data performs significantly better. When the variance of Gaussian distribution increases, the accuracy rate of both traditional EM and EM with unlabeled data decreases as shown in Figure 2.

An important point here is that the outliers in labeled data can hurt the performance. For example, in DS4, when the number of labeled data is less than 120, the EM without unlabeled data performs better accuracy than EM with unlabeled data. However, when the labeled data increases (the affect of outlier decreases), the EM with unlabeled data achieves better. The reason is, as Shahshahani et al. [18] point out, that although in theory the additional unlabeled samples should always improve the performance, in practice this might not always be true. As the unlabeled samples might contain outliers due to the deviation of the real world situations from the models that are assumed. Such outliers can hurt the performance.

In figure 3, we hold the number of labeled data constant as 100 (the class distribution is the same as DS2), and vary the number of unlabeled data in the horizontal axis. The experiment results show that more unlabeled data natually improve the accuracy rate. The exponential rate of convergence towards the limiting rate is evidenced by the approximate linear trend in the semilog Figure 3.

4.1.2 *Scalability Evaluation of B-EM*

In this experiement, we examine the scalability of B-EM as the size of input data set increases and compare the running time of B-EM with that of EM. The data sets in this evaluation have $d = 25$ attributes and has $k = 10$ class (the description of class distribution is the same as DS3). We generate the data sets with the size 100k, 200k, 500k, 1M and 5M records respectively. For all data sets, we fix the labeled records as 120.

B-EM was run with memory size of 1000 records while EM was not constrained to a limited RAM requirement. We note that for data set with 25 continuous attributes, the 5M records need 1G memory (ignoring any other RAM requirements). The bootstrap step M is 100.

Figure 4 shows the overall running times of the algorithms as the number of records in the data sets increases from 100k to 5M. As can be observed, the B-EM method scales well with the size of the data set, while the execution time of the standard EM gets to be impraticial for large sizes (more than 2 hours for 5M records). It is also worth pointing out that B-EM is running faster than the full in-memory EM algorithm for data sets that fit in memory: 100k, 200k and 500k records as the explaination is given in the algorithm section. Note the classification accuracy of B-EM and EM for this experiment is almost the same (the difference is less than 0.5%).

4.2 Corel Data

Our studies here have focused on building classifier through EM over the Corel images collection [15]. The 68,040 images are cataloged into broad categories. Four sets of features, color histogram, color histogram layout, color moments, and co-occurence texture are available online.

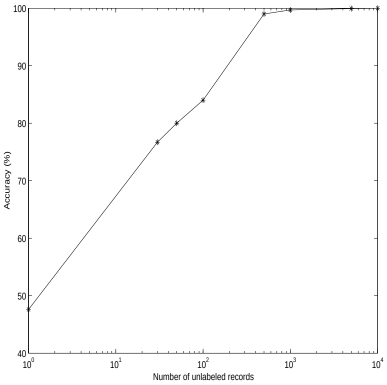

Figure 3: The effect of varying the number of unlabeled data. Classification accuracy is shown on synthethic data set with 100 labeled records, and varying amounts of unlabeled records

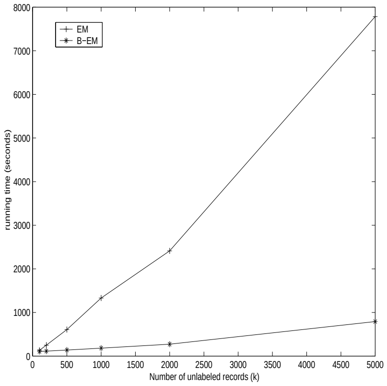

Figure 4: Execution time vs. number of records for EM and B-EM

In this experiment, we extract 6003 images and get 12 classes. The experiment is done over the combined feature set (41 features) of color histogram and color moment.

Figure 5 shows the effect of using EM with and without unlabeled images. EM with unlabled images performs better when only a small number of labeled images are available. When we have sufficient number of labeled images, as shown in right figure in Figure 5, the unlabeled images do not help improve the performance.

5. CONCLUSIONS

This paper investigate the question how unlabeled data may be used to improve the accuracy of classifier when few labeled data is available. This is an important question in many applications such as document categorization and image classification where the cost of labeling is very high and the hugh volume or unlabeled data is easily available. We then present a scalable classification algorithm, B-EM, to exploit the huge volume of unlabeled data while avoding I/O cost.

The theoretical model shows unlabeled data can be used to improve the accuracy of classifiers when 1) the probability distribution that generates the underlying data can be de-

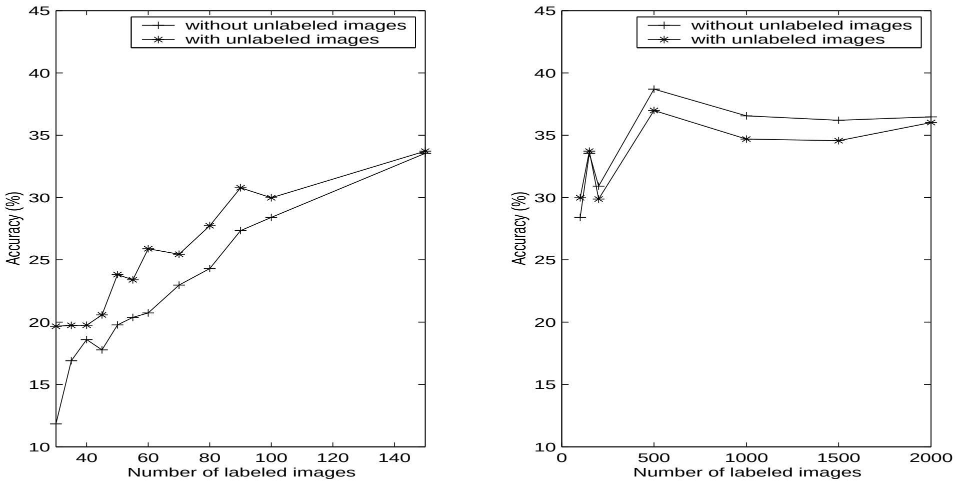

Figure 5: Classification accuracy on the corel data set (6003 images, 12 classes), both with and without unlabeled images

scribed as a mixture distribution and 2) there is a one-to-one correspondence between the components and class labels. However, the complexity of real-world applications will not be completely captured by statistical models and the real-world data is not totally consistent with the assumptions of the model. For example, we assume one Gaussian distribution over feature space for mamal in our experiment with Corel data set. This assumption is clearly not true. Our future work will investigate how to build mixture models over concept hierarchy.

Another interesting direction for future work with unlabeled data is how to build an incremental learning algorithm for stream data [1] where the unlabeled data is infinitely available. The incremental algorithm may expliot the unlabeled test data received in the testing phase to improve performance on the later test data.

6. ACKNOWLEDGMENTS

The authors would like to thank Michael Ortega-Binderberger for his help in providing the labels of Corel images.

7. REFERENCES

[1] S. Babu and J. Widom. Continuous queries over data streams. *SIGMOD Record*, 30(3), Sept 2001.

[2] P. S. Bradley, U. M. Fayyad, and C. A. Reina. Scaling clustering algorithms to large databases. In *Proceedings of the Fourth ACM KDD International Conference on Knowledge Discovery and Data Mining*, pages 9–15, August 1998.

[3] V. Castelli and T. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 6:105–111, 1995.

[4] V. Castelli and T. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameters. *IEEE Transaction on Information Theory*, 42(6), 1996.

[5] B. Chapmann and R. Tibshirani. *An Introduction to the Bootstrap. Monograph on Statistics and Applied Probability*. Chapman and Hall, 1993.

[6] P. Cheeseman and J. Stutz. Bayesian classification (autoclass): Theory and results. *Advances in Knowledge Discovery and Data Mining*, 1996.

[7] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[8] J. Gehrke, V. Ganti, R. Ramakrishnan, and W.-Y. Loh. Boat-optimistic decision tree construction. In *Proceedings of the SIGMOD Conference*, pages 169–180, 1999.

[9] Z. Ghaharmani and M. Jordan. Supervised learning from incomplete data via an em approach. *Advances in Neural Information Processing Systems 6*, 1994.

[10] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Morgan Kaufmann, 1999.

[11] M. James. *Classification Algorithms*. Wiley, 1985.

[12] R. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 4(22), 1987.

[13] M. Mehta, R. Agrawal, and J. Risanen. Sliq: A fast scalable classifier for data mining. In *Proceedings of the Fifth International Conference on Extending Database Technology*, March 1996.

[14] K. Nigam, A. Mccallum, S. Thrun, and T. Mitchel. Text classification from labeled and unlabeled documents. *Machine Learning*, 39(2/3):103–134, 2000.

[15] M. Ortega-Binderberger. Corel image features. http://kdd.ics.uci.edu/databases/CorelFeatures/CorelFeatures.html.

[16] J. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[17] J. Shafer, R. Agrawal, and M. Mehta. Sprint: A scalable parallel classifier for data mining. In *Proceedings of the 22nd VLDB Conference*, pages 544–555, 1996.

[18] B. Shanhshanhani and 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*, 32(5):1087–1095, 1994.