Semi-supervised Sequence Classification with HMMs

Shi Zhong
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431, USA
zhong@cse.fau.edu

Abstract

Using unlabeled data to help supervised learning has become an increasingly attractive methodology and proven to be effective in many applications. This paper applies semi-supervised classification algorithms, based on hidden Markov models (HMMs), to classify sequences. For model-based classification, semi-supervised learning amounts to using both labeled and unlabeled data to train model parameters. We examine three different strategies of using labeled and unlabeled data in the model training process. These strategies differ in how and when labeled and unlabeled data contribute in the whole model training process. Our experimental results on synthetic and real EEG time-series show that substantially improved classification accuracy can be achieved by these semi-supervised learning strategies.

Introduction

Learning with both labeled and unlabeled data has been studied with great interest recently. Though theoretical justification on the value of unlabeled data has not been promising (Castelli & Cover 1996; Zhang & Oles 2000), novel semi-supervised algorithms and successful applications are abundant (Blum & Mitchell 1998; Joachims 1999; Nigam *et al.* 2000; Wu & Huang 2000; Blum & Chawla 2001; Basu, Banerjee, & Mooney 2002). For example, it has been shown that unlabeled data can significantly improve the classification accuracy or information retrieval performance in applications such as text classification (Joachims 1999; Nigam *et al.* 2000), terrain classification (Guerrero-Curieses & Cid-Sueiro 2000), gesture recognition (Wu & Huang 2000), and content-based image retrieval (Dong & Bhanu 2003). Semi-supervised learning can also be viewed from another angle: labeled data used as feedbacks to help cluster unlabeled data. This leads to semi-supervised clustering; e.g., Basu, Banerjee, & Mooney (2002) studied the effectiveness of seeded k-means and constrained k-means as semi-supervised techniques for clustering text documents.

Semi-supervised learning has been motivated by many real-world problems. For example, text categorization can be tedious for a human—one has to read through the document and put it in an appropriate predefined category. Many recent works (Nigam *et al.* 2000; Nigam 2001) aim to reduce the number of text documents to be labeled by a human while achieving the same level of classification accuracy by exploiting information in unlabeled documents. In gene expression analysis, profound expert knowledge and costly biological experiments are often required to manually label the functional category of each gene. Semi-supervised learning methods can help reduce such efforts by automatically grouping unlabeled/unknown genes into meaningful categories based on only limited number of manually-classified genes.

With the same motivation, the intent of this paper is to classify sequence data with minimum human labeling efforts. Sequence classification problems are encountered in real applications such as robot motion control, biosignal analysis (Rezek & Roberts 2000; Zhong & Ghosh 2002), etc. The semi-supervised learning paradigm used in this paper is similar to the one in Nigam *et al.* (2000) and those in Basu, Banerjee, & Mooney (2002).

The contribution of this paper is the marriage of semi-supervised learning and hidden Markov models (HMMs) for sequence classification. To the best of our knowledge, an empirical study of semi-supervised learning for sequence classification has not been done before. To be precise, by "sequence", we refer to a sequence of numbers (discrete or continuous) at discrete time points. The time-series data type used in our experiments are a subtype of sequence data. Our methodology, however, applies to any sequence data that can be modeled by HMMs.

Instead of extracting a vector of features, we model sequences using HMMs, which have proven to be very effective in characterizing a wide variety of sequential data (Rabiner 1989; Rezek & Roberts 2000). For example, the real EEG time-series (used in our experiments) can be effectively classified using HMMs (Zhong & Ghosh 2002). We compare three different semi-supervised learning strategies of using labeled and unlabeled data to train HMMs and classify sequences. The effectiveness of these strategies are demonstrated by experiments on synthetic as well as real EEG time-series.

The organization of this paper is as follows. Semi-supervised sequence learning problem is formalized in the

the next section, followed by an introduction to HMMs. We then present three semi-supervised classification strategies and show improved classification results on two time-series datasets. Finally, we conclude this paper with remarks on related work and future work.

Semi-supervised Learning Problem

There are two types of data: labeled data $\mathcal{L} = \{o_i^{(l)}, y_i^{(l)}\}_{i=1}^{N_l}$ and unlabeled data $\mathcal{U} = \{o_i^{(u)}\}_{i=1}^{N_u}$, where N_l is the number of labeled data instances, N_u the number of unlabeled instances, and $\{y_i^{(l)}\}_{i=1}^{N_l}$ the class labels of labeled sequences. The goal is to label the unlabeled instances based on all data. This problem is general in that we do not know whether or not the class labels contained in labeled data is complete for unlabeled data. It is totally legal to discover new labels from unlabeled data, but can be very difficult. An extreme situation of the problem is the $N_l = 0$ case when one has to assign labels to all data. One then usually resorts to clustering algorithms, by which "similar" data instances are grouped together and assigned a group label. In this paper, we restrict ourselves to the scenario that labels contained in the labeled data are complete and we aim to assign these labels to each and every unlabeled data instance.

Our important assumption in this paper is that both labeled and unlabeled sequences are generated from the same set of models. That is, each class of sequences is modeled by an HMM λ and the likelihood $P(o|\lambda)$ used as a measure of how likely a sequence o is generated from the model λ. Next we briefly describe hidden Markov models.

Hidden Markov Models

HMMs have been heavily researched and used for the past several decades, especially in the speech recognition area (Rabiner 1989). A standard HMM model uses a discrete hidden state at time t to summarize all the information before t and thus the observation at any time only depends on the current hidden state. The hidden state sequence is a Markov chain. In this paper we use the simplest HMM, a univariate first order HMM, in which the observation is a scalar at any time and the state sequence is a first order Markov chain. Such an HMM unrolled over several time slices is shown in Fig. 1.

A standard HMM is usually denoted as a triplet $\lambda = (\pi, A, B)$. $\pi = \{\pi_i\}$ (where $\sum_i \pi_i = 1$) is the prior probability distribution of hidden states. $A = \{a_{ij}\}$ (where $\sum_j a_{ij} = 1$) is the transition probability distribution between hidden states. For discrete observation case, the observation distribution is $B = \{b_j(k)\}$ (where $\sum_k b_j(k) = 1$). For continuous observation case, the observation distribution is usually modeled by a mixture of Gaussians

$$b_j(o) = \sum_l c_{jl} \mathcal{N}[o, \mu_{jl}, U_{jl}] \,, \qquad (1)$$

where $\sum_l c_{jl} = 1$, o is the observation vector being modeled, c_{jl} the mixture weight, μ_{jl} the mean vector of the m-th mixture, U_{jl} the covariance matrix of the l-th mixture for state j and $\mathcal{N}$ is the Gaussian density function.

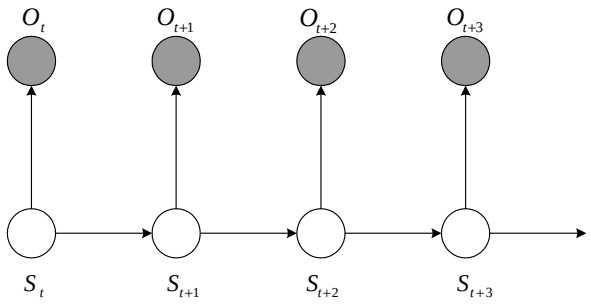


Figure 1: A first order HMM model. The empty circles are hidden states and the shaded ones observations.

Recently, HMMs have been extended to solve various time-series and sequence data analysis problems, such as protein sequence modeling (Eddy 1998) and biosignal analysis (Rezek & Roberts 2000).

Semi-supervised HMM-based Classification (SSHC)

It is worth noting that semi-supervised model-based classification is closely related to model-based partitional clustering (Zhong & Ghosh 2003), which involves two basic steps: (a) In a data assignment step each data instance is assigned to one or more model(s); (b) In a model estimation step, one estimates the parameters of each model using the data instances assigned to it. In this paper, we consider "hard" assignment, where each data instance is assigned to only one model/cluster, which results in a k-means type algorithm that has been used by many researchers (Dermatas & Kokkinakis 1996; Smyth 1997; Law & Kwok 2000; Li & Biswas 2002).

By modifying the HMM-based k-means to accommodate labeled sequences, we can get semi-supervised classification algorithms. The modification basically amounts to using labeled data to initialize HMMs and/or to constrain the estimation of HMMs. The next section presents three different versions of semi-supervised HMM-based sequence classification algorithms. The first two versions can be seen as HMM-based extensions of the constrained k-means and seeded k-means algorithms proposed in Basu, Banerjee, & Mooney (2002).

Semi-supervised Algorithms

The first version, we call *SSHC-1*, is shown in Fig. 2. It is also the most straightforward combination of HMM-based k-means with supervised training—first initializing HMMs using labeled sequences and then iterating between two steps: labeling unlabeled sequences and updating models using both the original labeled sequences and the newly-labeled sequences. In this version, the labeled sequences are used to constrain the whole training process of HMMs.

We call Step 1 of the SSHC-1 algorithm *supervised* step and Step 2&3 *semi-supervised* steps. The other two versions

Algorithm: SSHC-1

Input: A set of N_l labeled sequences $O^{(l)} = \{o_1^{(l)}, ..., o_{N_l}^{(l)}\}$ with labels $Y^{(l)} = \{y_1^{(l)}, ...y_{N_l}^{(l)}\}$, N_u unlabeled sequences $O^{(u)} = \{o_1^{(u)}, ..., o_{N_u}^{(u)}\}$, HMM structure $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_K\}$.

Output: Trained model parameters $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_K\}$ and a partition of the unlabeled sequences given by the identity vector $Y^{(u)} = \{y_1^{(u)}, ...y_{N_u}^{(u)}\}$, $y_i^{(u)} \in \{1, ..., K\}$.

Steps:

1. Supervised training: for each class j, let $O_j^{(l)} = \{o_i^{(l)} | y_i^{(l)} = j\}$, an HMM model is trained as
$$\lambda_j = \max_{\lambda} \sum_{o \in O_j^{(l)}} \log p(o|\lambda) ;$$
2. Data assignment: for each unlabeled sequence $o_i^{(u)}$, set $y_i^{(u)} = \arg\max_j \log p(o_i^{(u)}|\lambda_j)$;
3. Model estimation: let $O_j^{(u)} = \{o_i^{(u)} | y_i^{(u)} = j\}$ and $O_j = O_j^{(l)} \cup O_j^{(u)}$, the parameters of model λ_j is re-estimated as $\lambda_j = \max_{\lambda} \sum_{o \in O_j} \log p(o|\lambda)$;
4. Stop if $Y^{(u)}$ does not change, otherwise go back to Step 2.

Figure 2: Semi-supervised HMM-based classification - 1.

of the SSHC algorithm differ from the first one in just the *semi-supervised* steps. Fig. 3 shows only the Step 2&3 for SSHC-2 algorithm, which frees labeled sequences after the supervised step and consequently a labeled sequence may be assigned a different label in the later semi-supervised iterative process. This is based on the idea that there may be noise (in the labeled sequences) that may prevent the HMMs from fitting (and classifying) unlabeled sequences better.

For SSHC-3, shown in Fig. 4 (again, only Step 2&3), we use the trained HMMs from supervised step as a starting point for clustering on unlabeled sequences. That is, we use only unlabeled sequences in semi-supervised iterative process. Intuitively, this one should underperform SSHC-1 and SSHC-2 since it uses less information (unlabeled data only) after the supervised step. We include it here, however, for completeness. The words in boldface in Fig. 3 & 4 highlight the key differences of these algorithms.

Algorithm: SSHC-2 (Step 2&3)

Steps:

2. Data assignment: for every (**labeled and unlabeled**) sequence o_i, set $y_i = \arg\max_j \log p(o_i|\lambda_j)$;
3. Model estimation: let $O_j = \{o_i^{(l)} | y_i^{(l)} = j\} \cup \{o_i^{(u)} | y_i^{(u)} = j\}$, the parameters of model λ_j is re-estimated as $\lambda_j = \max_{\lambda} \sum_{o \in O_j} \log p(o|\lambda)$;

Figure 3: Semi-supervised HMM-based classification - 2.

Algorithm: SSHC-3 (Step 2&3)

Steps:

2. Data assignment: for each unlabeled sequence $o_i^{(u)}$, set $y_i^{(u)} = \arg\max_j \log p(o_i^{(u)}|\lambda_j)$;
3. Model estimation: let $O_j^{(u)} = \{o_i^{(u)} | y_i^{(u)} = j\}$, **using only unlabeled sequences**, the parameters of model λ_j is re-estimated as $\lambda_j = \max_{\lambda} \sum_{o \in O_j^{(u)}} \log p(o|\lambda)$;

Figure 4: Semi-supervised HMM-based classification - 3.

We do not have enough space to give detailed analysis of time complexities for these algorithms but it can be easily verified that the complexity for SSHC-1 and SSHC-2 is $O(KMM_1NT^2)$, where M is the number of semi-supervised iterations, M_1 the number of iterations used for maximum likelihood estimation of an HMM model, and T the sequence length. Note the complexity of training an HMM is just $O(M_1NT^2)$. SSHC-3 has slightly lower complexity since it uses only N_u unlabeled sequences in the semi-supervised steps.

Experimental Study

Datasets

We experiment on two datasets—a synthetic HMM-generated dataset and a real EEG dataset. The synthetic dataset is the same as the one used by (Smyth 1997). 40 sequences of length $T(= 200)$ are generated from two continuous HMM models (HMM1 and HMM2), 20 from each. The number of hidden states is 2 from both models. The prior and observation parameters for HMM1 and HMM2 are the same. The prior is uniform and the observation distribution is univariate Gaussian with mean $\mu = 3$ and variance $\sigma^2 = 1$ for hidden state 1, and mean $\mu = 0$ and variance $\sigma^2 = 1$ for hidden state 2. The state transition parameters of HMM1 and HMM2 are $A_1 = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$ and $A_2 = \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$, respectively.

The real dataset is a small EEG dataset downloaded from UCI KDD Archive web site (http://kdd.ics.uci.edu/). It contains measurements from 64 electrodes on a human scalp. In our experiment we only extract data from one electrode (F4) and model it with a univariate HMM. In the future we intend to model multiple electrodes with multivariate HMMs. There are 20 measurements from two subjects, a control subject and an alcoholic subject, 10 from each. The measurement is sampled at 256Hz for 1 second, producing a sequence length of 256. The goal is to classify the subject as normal or alcoholic based on the EEG time-series data.

Geva & Kerem (1998) clustered EEG time-series using weighted fuzzy k-means algorithm on extracted feature vector space. But expert knowledge is required to extract good features. In this paper no feature extraction is needed; the raw time-series are modeled with HMMs. EEG signal is believed to be highly correlated with the sleep stages of brain

cell. The number of sleep stages is about 5 or 6 according to Geva & Kerem. Therefore, the correct number of hidden states is assumed to be 5 or 6.

Experimental Setting

A few details on maximum likelihood training of HMMs are worth mentioning here. Juang, Levinson, & Sondhi (1986) pointed out that using mixture of Gaussians as the observation model of HMM sometimes results in singularity problems during training. They suggest solving the problem by re-training from a different initialization, which is the way we deal with singularity problem in our experiments. Rabiner *et al.* (1985) observed, through empirical study, that accurately estimating the means of Gaussians is essential to learning good models for continuous HMMs. We used the standard k-means algorithm to locate the means for the observation Gaussian distributions, following the approach used in Smyth (1997). Random initialization is used for other parameters.

In addition to the initialization scheme used above, we try to exploit the labeled information by rejecting random initializations that fail to generate models better than baseline model (that is, the random guess model with 50% classification accuracy). Our experiments show that this is useful in improving classification accuracy by getting rid of some bad starting points.

For the EEG data, we scale all values (the value of every labeled and unlabeled sequences at every time slice) to be with $[-5, 5]$ to avoid severe mismatch between data and initial random models.

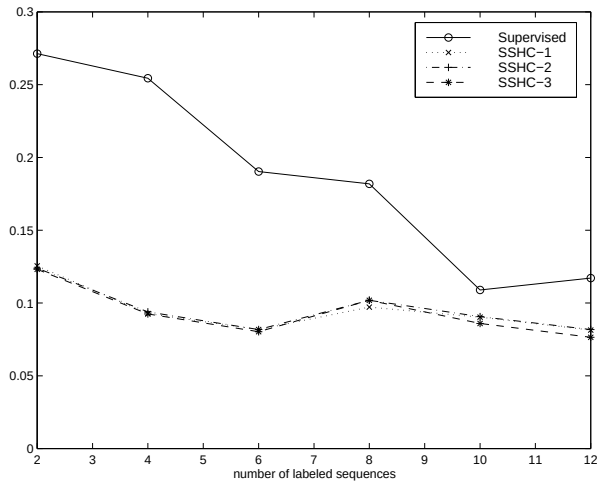
For supervised classification, we need to specify part of the data as labeled data. To see how the number of labeled sequences affect the performance of semi-supervised learning, we experiment on different number of labeled sequences. We vary the number from 2 to 12 for the synthetic data and from 2 to 10 for the real EEG data. Given N_l (number of labeled sequences), we simply pick the first $N_l/2$ samples from class 1 and class 2, respectively.

We run each experiment 20 times and report the average classification errors. We try two different number of hidden states for each data, 2 and 5 for the synthetic data and 5 and 8 for the EEG data, to see the effect of model complexity.

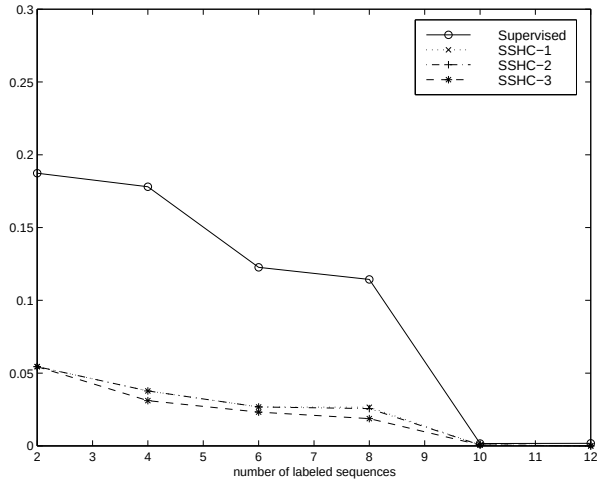
Results Analysis

Fig. 5 and 6 show the classification results on the synthetic data and the EEG data, respectively. In each figure, four curves of misclassification rate (y-axis) vs. number of labeled sequences (x-axis) are shown, corresponding to supervised method and the three semi-supervised algorithms discussed above, respectively.

First of all, we can see that SSHC-1 and SSHC-2 have almost the same performance in all situations. This is plausible since these two algorithms both use all data, i.e., see the same amount of information for semi-supervised training. It may also indicate that the noise contained in labeled data is not an important factor in most cases. The SSHC-3 algorithm, which uses only unlabeled sequences after supervised step, fares worse than SSHC-1 and 2 in some cases,



(a)

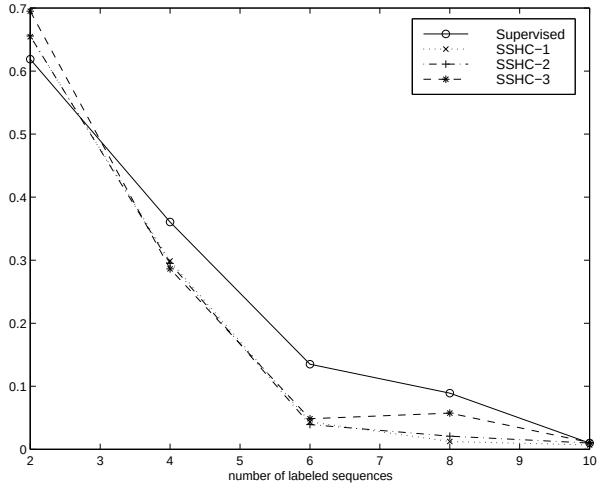


(b)

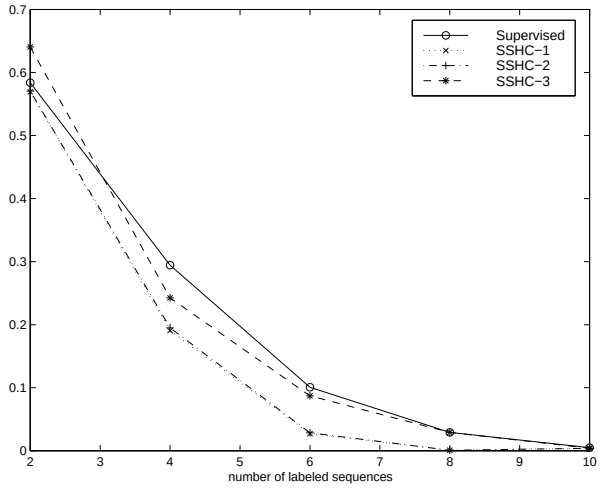
Figure 5: Classification error results on synthetic data using HMMs with: (a) 2 hidden states and (b) 5 hidden states.

especially for the EEG dataset. The results suggest that labeled sequences contribute in the whole process—both supervised step and semi-supervised step.

Now let us look at SSHC-1 only. The results clearly show that unlabeled sequences improve the classification accuracy significantly in most situations. The only case where SSHC-1 does slightly worse than supervised learning is when using two labeled sequences in the EEG case. Notice that the supervised learning produces more than 60% error that is worse than baseline error. This indicates that the two labeled sequences happen to be bad training instances. In some other cases, when the number of labeled sequences exceeds certain amount, SSHC-1 performs approximately the same as supervised learning. This is because the amount of labeled sequences has become large enough to train accurate HMM models and additional unlabeled sequences provide little extra help. As shown in the graphs, the gap between



(a)



(b)

Figure 6: Classification error results on EEG data using HMMs with: (a) 5 hidden states and (b) 8 hidden states.

SSHC-1 and supervised learning shrinks as the number of labeled sequences increases.

Comparing (a) with (b) in both figures, one can see that increasing the complexity (number of hidden states) of HMM models leads to (significantly) lower classification error, even though more hidden states should lead to more local maxima and the numbers exceed "correct" values (2 for synthetic data and 5 for EEG data). We suspect the reason is that, since labeled sequences provides a good starting point, the benefit of getting more discriminating power from more complex model outweigh the negative effect of having more local maxima.

Related Work

Various semi-supervised algorithms have been proposed. Blum & Mitchell (1998) introduced co-training algorithm for learning from labeled and unlabeled data. They assume that there exist two independent sets of features and either set can confidently predict the class labels of unlabeled data. Their method performs well for real text data despite the strong assumption. Joachims (1999) proposed transductive SVM—a method to incorporate unlabeled data into the formulation of a SVM classifier. The Basic idea is to keep unlabeled data far away from the decision boundary in addition to trying to maximize the decision margin for labeled data. Good performance of the transductive SVM has been shown on text classification problems.

Guerrero-Curieses & Cid-Sueiro (2000) used labeled data to minimize the cost function of a classifier and unlabeled data to minimize some corresponding entropy measure. Their method is applicable only when the classifier outputs class probabilities. In their formulation, minimizing entropy is equivalent to minimizing uncertainty of unlabeled data, which has the same flavor of forcing unlabeled data to be away from the most uncertain region (i.e. decision boundary) as transductive SVM. Blum & Chawla (2001) proposed to use graph mincut method to do semi-supervised learning. Their method perform comparably with other (e.g. EM) method. Similarity measure between any two data instances is needed. The computation effort to construct the graph seems to be high.

Our work is most similar to Nigam *et al.* (2000), who employed naïve Bayes model with EM algorithm for classifying text documents with labeled and unlabeled data. They show that the EM approach can improve the classification performance substantially on some data while hurting accuracy on other ones. They attribute the negative experiences to serious mismatch between the naïve Bayes model and the data. In this paper, we used k-means (instead of EM) for its simplicity.

There have been some studies on the relative value of labeled and unlabeled data for classification. Unfortunately, results from these studies suggest little value with unlabeled data relative to labeled data. Castelli & Cover (1996) proved that labeled examples are exponentially more valuable than unlabeled examples in pattern recognition task. But they make very strong assumptions that the input distribution is known completely and that all class-conditional distributions can be learned from unlabeled data only. These assumptions usually do not hold in reality. In a recent study, Zhang & Oles (2000) also questioned the usefulness of transductive SVMs.

Finally, Seeger (2001) provides a good summary of recent development in semi-supervised learning with labeled and unlabeled data.

Concluding Remarks

For the time-series classification problems studied in this paper, we have shown that unlabeled sequences can improve classification accuracy significantly when the supervised learning performance is reasonably good. And we observed in our experiments that more complex models further improve the classification accuracy. This is not necessarily true in general; model selection techniques can be used to choose a good level of complexity.

In the future, we plan to run more experiments on larger EEG time-series datasets and see whether our observations in this paper are still true. More stable training algorithms (to eliminate the singularity problem) such as MAP (maximum *a posteriori*) learning (Gauvain & Lee 1994) with appropriate priors can be employed.

Some other directions in which we can proceed include:

- Incrementally labeling the unlabeled data (Nigam & Ghani 2000) may improve the semi-supervised learning results because of its less greedy optimization approach. Currently, all the unlabeled data are "labeled" (for model training) in batch at every iteration.
- By integrating model selection method into our algorithm, we shall investigate the estimation of model structure parameters such as the number of clusters (models) and the number of hidden states for HMM models. The interaction between semi-supervised learning and model selection also deserves future investigation.

References

Basu, S.; Banerjee, A.; and Mooney, R. 2002. Semi-supervised clustering by seeding. In *Proc. 19th Int. Conf. Machine Learning*, 19–26.

Blum, A., and Chawla, S. 2001. Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th Int. Conf. Machine Learning*, 19–26.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *The 11th Annual Conf. Computational Learning Theory*, 92–100.

Castelli, V., and Cover, T. M. 1996. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Trans. Information Theory* 42(6):2102–2117.

Dermatas, E., and Kokkinakis, G. 1996. Algorithm for clustering continuous density HMM by recognition error. *IEEE Trans. Speech and Audio Processing* 4(3):231–234.

Dong, A., and Bhanu, B. 2003. A new semi-supervised em algorithm for image retrieval. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, volume 2, 662–667.

Eddy, S. R. 1998. Profile hidden Markov models. *Bioinformatics* 14:755–763.

Gauvain, J.-L., and Lee, C.-H. 1994. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. Speech and Audio Processing* 2(2):291–298.

Geva, A. B., and Kerem, D. H. 1998. Brain state identification and forecasting of acute pathology using unsupervised fuzzy clustering of EEG temporal patterns. In Teodorescu, H.-N.; Kandel, A.; and Jain, L. C., eds., *Fuzzy and Neuro-Fuzzy Systems in Medicine*. CRC Press. chapter 3, 57–93.

Guerrero-Curieses, A., and Cid-Sueiro, J. 2000. An entropy minimization principle for semi-supervised terrain classification. In *Proc. IEEE Int. Conf. Image Processing*, volume 3, 312–315.

Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *Proc. 16th Int. Conf. Machine Learning*, 200–209.

Juang, B.-H.; Levinson, S. E.; and Sondhi, M. M. 1986. Maximum likelihood estimation for multivariate mixture observations of Markov chains. *IEEE Trans. Information Theory* 32(2):307–309.

Law, M. H., and Kwok, J. T. 2000. Rival penalized competitive learning for model-based sequence clustering. In *Proc. IEEE Int. Conf. Pattern Recognition*, 195–198.

Li, C., and Biswas, G. 2002. Applying the hidden Markov model methodology for unsupervised learning of temporal data. *International Journal of Knowledge-based Intelligent Engineering Systems* 6(3):152–160.

Nigam, K., and Ghani, R. 2000. Understanding the behavior of co-training. In *KDD Workshop on Text Mining*.

Nigam, K.; Mccallum, A.; Thrun, S.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning* 39(2/3):103–134.

Nigam, K. 2001. *Using Unlabeled Data to Improve Text Classification*. Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University.

Rabiner, L. R.; Juang, B.-H.; Levinson, S. E.; and Sondhi, M. M. 1985. Some properties of continuous hidden Markov model representations. *AT&T Technical Journal* 64(6):1251–1269.

Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of The IEEE* 77(2):257–286.

Rezek, I., and Roberts, S. J. 2000. Estimation of coupled hidden Markov models with application to biosignal interaction modeling. In *Proc. IEEE Int. Conf. Neural Network for Signal Processing*, volume 2, 804–813.

Seeger, M. 2001. Learning with labeled and unlabeled data. Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh.

Smyth, P. 1997. Clustering sequences with hidden Markov models. In Mozer, M. C.; Jordan, M. I.; and Petsche, T., eds., *Advances in Neural Information Processing Systems 9*, 648–654. MIT Press.

Wu, Y., and Huang, T. S. 2000. Self-supervised learning for visual tracking and recognition of human hand. In *Proc. 17th National Conference on Artificial Intelligence*, 243–248.

Zhang, T., and Oles, F. 2000. A probabilistic analysis on the value of unlabeled data for classification problems. In *Proc. 17th Int. Conf. Machine Learning*, 1191–1198.

Zhong, S., and Ghosh, J. 2002. HMMs and coupled HMMs for multi-channel EEG classification. In *Proc. IEEE Int. Joint Conf. Neural Networks*, 1154–1159.

Zhong, S., and Ghosh, J. 2003. A unified framework for model-based clustering. *Journal of Machine Learning Research* 4:1001–1037.