A Comparison of Discriminative EM-Based Semi-Supervised Learning algorithms on Agreement/Disagreement classification

Sangyun Hahn
Computer Science and Engineering Department
University of Washington
Seattle, WA 98195, USA
syhahn@cs.washington.edu

Mari Ostendorf
Electrical Engineering Department
University of Washington
Seattle, WA 98195, USA
mo@ee.washington.edu

Abstract

Recently, semi-supervised learning has been an active research topic in the natural language processing community, to save effort in hand-labeling for data-driven learning and to exploit a large amount of readily available unlabeled text. In this paper, we apply EM-based semi-supervised learning algorithms such as traditional EM, co-EM, and cross validation EM to the task of agreement/disagreement classification of multi-party conversational speech, using discriminative models such as support vector machines and multi-layer perceptrons. We experimentally compare and discuss their advantages and weaknesses when used with different amounts of unlabeled data.

1 Introduction

A data-driven approach to natural language processing problems typically requires a large amount of data, but it is often very expensive or impossible to obtain a sufficient amount of labeled data. To exploit a large amount of easily available unlabeled data as well as labeled data, various semi-supervised learning algorithms have been proposed [**?**]. Among them, the Expectation-Maximization (EM) algorithm [**?**] is one of the oldest and most widely used algorithms in semi-supervised learning due to its applicability to a wide range of problems. Co-training [**?**] is another popular semi-supervised learning algorithm, which exploits two independent views of the same data to explore different data regions. Both algorithms have advantages and weaknesses; for example, EM can find a local maximum or overfit to training data, and co-training may not perform well when splitting features into independent sets is not possible [**?**, **?**, **?**]. Thus, there has been some effort to combine these two approaches to improve their classification performance [**?**, **?**]. In this paper, we compare the performance of EM and its variants, including co-EM [**?**, **?**] and cross validation EM (CVEM) [**?**]. Performance is compared for large vs. small amounts of unlabeled data.

For EM, typically a generative model such as Gaussian mixtures or Naive-Bayes is used to learn class-conditional probability distributions $p(x|y, \theta)$ from a given data set, where x is the observation and y is the class. In general, discriminative models are preferred for a classification task, because they often outperform generative models by directly learning the class posterior $p(y|x, \theta)$ used in the decision function [**?**, **?**, **?**]. In our experiments, we used discriminative models (support vector machines (SVMs) or multi-layer perceptrons (MLPs)) in these variants of EM algorithms, adapting Brefeld's approach [**?**].

Our experiments with semi-supervised learning were performed on the task of identifying agreements and disagreements in multi-party conversational speech. Originally proposed as a 4-way classification task of "positive" (agreement), "negative" (disagreement), "backchannel", and "other",

these labels represent a simple type of "speech act" that can be important for understanding the interaction between speakers, or for automatically summarizing or browsing the contents of a meeting. This problem was previously studied [**?**, **?**, **?**] using the ICSI meeting recording corpus [**?**]. The class distribution in the labeled data set is somewhat skewed (over 60% of data from the majority class, and only 5% from the disagreement class).

This paper is organized as follows. In section 2, we briefly describe existing EM-based and related semi-supervised learning algorithms relevant to our study, with respective advantages and weaknesses noted in the literature. In section 3, we provide a new variant and discuss implementation details of the algorithms used in our experiments. Then, we present the results and important findings from experiments in section 4, and conclusions are provided in section 5.

2 Background

Given labeled data $D_l = \{(x_1, y_1), \ldots, (x_L, y_L)\}$ and unlabeled data $D_u = \{x_{L+1}, \ldots, x_{L+U}\}$ where $x_i \in \mathcal{R}^d$ and $y_i \in \{1, \ldots, K\}$, the goal of semi-supervised learning is to find a function $f : \mathcal{R}^d \rightarrow \{1, \ldots, K\}$ that obtains the smallest classification error possible on any given new data points as well as on the unlabeled data. Expectation-Maximization (EM) [**?**] is one of the oldest semi-supervised learning algorithms, which treats missing labels as hidden variables and estimates the expected joint likelihood $p(x, y|\theta)$ using an iterative algorithm. In the E-step, the expected statistics for hidden variables are computed based on the given model, and in the M-step, model parameters θ are estimated from the expected statistics provided in the E-step. By repeating these two steps, the EM algorithm gradually increases the likelihood of the training data. Typically, it uses generative models such as Naive-Bayes or Gaussian mixture models, and unlabeled data is guaranteed to improve classification accuracy as long as the model assumption is correct [**?**]. However, the performance of the resulting classifier can be hurt by local minima, overfitting to the training data, incorrect model assumptions, etc [**?**, **?**].

Another prominent semi-supervised learning algorithm, co-training [**?**], exploits two different views on a dataset to obtain a classification model. It assumes that features can be split into two sets conditionally independent of each other, given their class labels, and the true model can be trained using each of them alone. First, a classifier is trained with feature set A, using labeled data only. Then, the classifier is used to label unlabeled data. It selects data points labeled with the highest confidence and adds them to the training set. Next another classifier is trained with feature set B, and the above process is repeated until there are no more data points to be added to the training data set. Nigam and Ghani [**?**] pointed out that co-training takes advantage of multi-view learning and selection of data points labeled with high confidence, and along these two axes, other semi-supervised learning algorithms can be placed: self-training with single view learning and selection of data, EM with single view learning and no selection, and co-EM with multi-view learning and no selection. They compared classification performance of these algorithms and empirically showed that multi-view learning helps to improve classification accuracy if its two assumptions are correct. However, others [**?**, **?**] argue that in some real world problems, it is not possible to split features into two independent sets, and in such cases, it is not easy to benefit from co-training, if not impossible. Goldman *et al.* [**?**] proposed to use two different learning algorithms to obtain different classification models in place of using two independent feature sets. This enables us to apply co-training to a wider range of problems without being restricted by independence assumptions; thus, we adopted the multi-classifier approach for our multi-view learning experiments.

Co-EM is a natural extension of EM to multi-view learning. It is less sensitive to being stuck in local maxima than EM, by exploiting two different views of the data [**?**, **?**]. Brefeld *et al.* also extended SVMs to be used in EM and co-EM to directly model class posterior probabilities $P(y|x)$ and to incorporate the probability weights in learning a new model. First, an SVM classifier was trained using labeled data. Next, this classifier is used to label the unlabeled data, providing class posterior probabilities for the entire data set. Then, a new SVM classifier was trained with the posterior probabilities as weights for the corresponding examples. For co-EM, two independent feature sets were used to train SVM classifiers, and they were used to find class posteriors for the other view. Their experiments were limited to binary classifications, but their results were quite promising.

To address the problem of overfitting in the EM algorithm, recently, cross validation EM (CVEM) was proposed by Shinozaki and Ostendorf [**?**], and it was successfully applied to HMM training

for large vocabulary Mandarin speech recognition. In CVEM, K models are trained using different partitions of the dataset in the M-step, and each model computes sufficient statistics for the subset not used for training the model. This is similar to K-fold cross validation techniques used for hyperparameter tuning, but it impacts all parameters of the model and is repeated with every iteration of EM. By using different data in the E- and M-steps, and learning multiple models using redundant data, it achieves reduced generalization error. Although EM is guaranteed to increase likelihood of the training data at each iteration, it is possible that the likelihood of the test data may go down. However, since CVEM computes the sufficient statistics for each group of data using a model which was estimated from different groups of data, the likelihood computed from these sufficient statistics follows a similar trend as the likelihood of the test set. This makes it possible to avoid overfitting and provides a way of detecting a stopping point.

3 EM-based semi-supervised learning algorithms

We explore four variations of the EM algorithm for semi-supervised learning: with vs. without co-training, and with vs. without the CV extensions. We use two types of discriminative models (SVM and MLP) to compute the class probability for unlabeled data. The different approaches are described below with an outline of steps in each of the alternatives for easy comparison.

In unsupervised EM, model parameters are initialized randomly, and hidden variables are unobserved for the full data set. In semi-supervised EM, the parameters are initialized based on labeled data, and for this subset of the full data, the "hidden" variables are observed. This means that the class posteriors are deterministic (1/0) and not updated in the E-step. All data, both labeled and unlabeled, is used in the M-step. The use of class-posteriors to weight the unlabeled data in the M-step means that the labeled data will effectively have more weight, but if there is a large amount of unlabeled data then it can dominate the new estimates. For that reason, it has been proposed to use an increasing constant C to control the effect of unlabeled data over iterations [**?**]. In our initial experiments, we obtained much better results without it, so it is not used in the work reported here, though further experiments with a different weight increase schedule might lead to improvement.

For SVMs, we use the LIBSVM library [**?**], which implements Platt's [**?**] method for binary SVM classifiers and multiclass probability estimate described in [**?**] for multiclass classification.[1] To train an SVM classifier using class probabilities for unlabeled data, they derived and solved an optimization problem associating weights with each unlabeled data point, and used the difference between the probabilities of two classes as its weight. We implemented this for multi-class classification. In LIBSVM, pairwise binary classifies are learned first and their outputs are combined to produce the final multi-class class probabilities. So we modified the implementation to use different weights for each binary classifier. The weight for unlabeled example x for the binary classifier that separates the most likely class i from another class j is set to $w^x_{i,j} = p(i|x) - p(j|x)$.

To incorporate an MLP in EM, we use the soft-max output function and the cross-entropy error measure in backpropagation learning to obtain the class probability model. The class posterior probabilities for unlabeled data, obtained from a previously trained classifier, were used to adjust the rate of updating network weights for their corresponding unlabeled examples during training iterations. That is, a larger weight for an unlabeled example results in a larger change in network weights. For the labeled examples, we set their weights to 1 in all iterations, so that we can fully take advantage of these labeled examples.

EM [?]

1. Train f_0 on labeled data D_l.
2. For $i = 1, \ldots, T$:
   - *E-Step:* Estimate class probabilities $\hat{p}(y|x_j)$ for $x_j \in D_u$, based on predictor f_{i-1}.
   - *M-Step:* Train f_i using D_l and D_u with the $\hat{p}(y|x_j)$ for $x_j \in D_u$.

---

[1] This approach is somewhat different from the approach in [**?**], where the class probability of an SVM output using Gaussian distribution, but thought to provide better probability estimates.

For co-EM, we use two different learning algorithms (SVM and MLP) in parallel, instead of two independent sets of features, as in [**?**]. Each classifier alternately labels unlabeled data probabilistically, and those labels are used to train the other classifier. In the final step, either of the the outputs of two classifiers can be used as the final classification or, as in this work, the outputs of the two classifiers can be combined. Here, the product rule outperformed interpolation.

---

co-EM [?]

1. Train f_0^2 on labeled data D_l with features V_2.
2. For $i = 1, \ldots, T$; For $v = 1, 2$:
   - *E-Step:* Estimate $\hat{p}^v(y|x_j)$ for $x_j \in D_u$ based on peer predictor $f_{i-1}^{\overline{v}}$.
   - *M-Step:* Train f_i^v using features V_v, data D_l and D_u with $\hat{p}^v(y|x_j)$ for $x_j \in D_u$.
3. Return: $f_T = \sqrt{(}f_T^1 \times f_T^2)$.

To implement CVEM, we again use either SVM or MLP as the underlying model. The labeled and unlabeled data are divided into K groups, and $K - 1$ groups are used to train a model which labeled the unlabeled data in the other one group probabilistically. In this way, K different models are trained at each iteration, and each of the K groups is relabeled using one of them.

CVEM [?]

1. Train f_0 using D_l;
   initialize $f_0^k = f_0$ for $k = 1, \ldots, K$
2. Split $D_l \cup D_u$ into $D_1, \ldots, D_K$
3. For $i = 1, \ldots, T$
   - *E-step*: for $k = 1, \ldots, K$:
     Estimate $\hat{p}^k(y|x_j)$ for unlabeled data $x_j \in D_k$, based on predictor f_{i-1}^k
   - *M-step:* for $k = 1, \ldots, K$:
     Train f_i^k using D_l for all $l \neq k$, with probability $\hat{p}^l(y|x_j)$ for unlabeled $x_j \in D_l$.
4. Train f_{T+1} using $D_1, \ldots, D_K$ with probabilities $\hat{p}^k(y|x_j)$ for unlabeled x_j.

Finally, as a natural extension of CVEM, we implemented co-CVEM which exploits both multi-view learning and cross validation techniques. Implementation details are the same as co-EM and CVEM.

---

co-CVEM.

1. Train f_0^2 on labeled data D_l with features V_2; initialize $f_0^{k2} = f_0^2$ for $k = 1, \ldots, K$
2. Split $D_l \cup D_u$ into $D_1, \ldots, D_K$
3. For $i = 1, \ldots, T$; For $v = 1, 2$:
   - *E-step:* For $k = 1, \ldots, K$:
     Estimate $\hat{p}^{k\overline{v}}(y|x_j)$ for unlabeled data $x_j \in D_k$, based on predictor $f_{i-1}^{k\overline{v}}$
   - *M-step:* For $k = 1, \ldots, K$:
     Train f_i^{kv} using D_l for all $l \neq k$, with features V_v, and probability $\hat{p}^{lv}(y|x_j)$ for unlabeled x_j.
4. For v=1,2: Train f_{T+1}^v using $D_1, \ldots, D_K$ with probabilities $\hat{p}^{kv}(y|x_j)$ for unlabeled $x_j \in D_k$.
5. Return: $f_{T+1} == \sqrt{(}f_T^1 \times f_T^2)$.

Note that in both co-EM and CVEM, unlike EM, likelihood is not guaranteed to increase monotonically. In addition, the use of discriminative models makes it difficult to compute exact log likelihood, since $p(y|x)$ is optimized in the maximization step rather than $p(x, y)$ (or $E[\log p(x, y)|x]$).

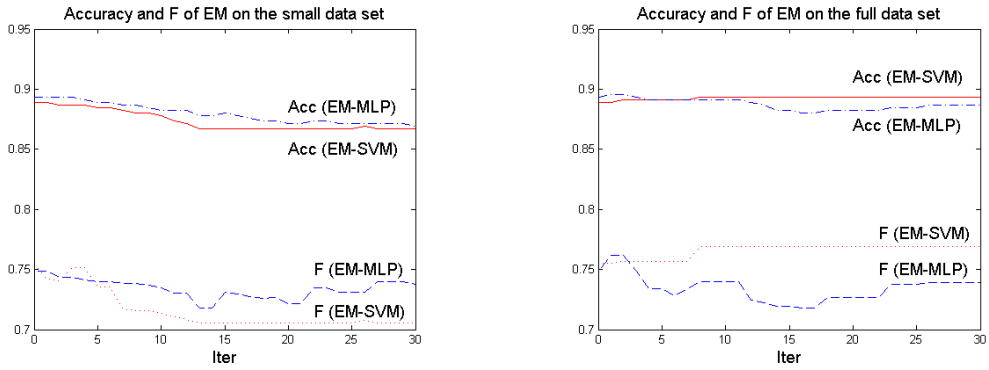

Figure 1: Accuracy and F of EM using SVM and MLP on the two ICSI data sets

4 Experiments and Results

We conducted experiments on the small dataset studied in previous work [**?**, **?**, **?**] and the entire ICSI meeting corpus. The small dataset contains 1803 labeled utterances taken from the first 15 minutes of 4 meetings, and 8101 unlabeled utterances from 7 meetings. Two thirds of labeled data are of the "other" class and only 6% are of the disagreement class. The test set comprised 451 labeled utterances from one meeting; the remaining labeled and unlabeled data was used for training as in previous work. We also use the rest of the ICSI meeting corpus (82850 spurts total) as unlabeled data to compare the effect of the amount of unlabeled data. Since "backchannel" is often used to encourage the speaker to continue, we grouped backchannel and agreements into one class for scoring purposes (as in previous work), resulting in a 3-class recognition problem. From the transcription for each utterance, we extracted such simple features as the number of words in an utterance, the number of keywords associated with the "positive" and "negative" classes, an estimation of class by keywords, and language model features such as perplexity and likelihood of the word sequence computed by language models for each of the four classes. These features are the same as those used in the experiments of [**?**]. In order to look for overtraining effects, we ran 30 iterations for each of the algorithms. Since the class distribution of this data is imbalanced, and the majority class is least important, here we present F score with or without accuracy in our results.

Figure 1 shows the accuracy and F scores over 30 iterations by EM using different underlying classifiers on the small and full ICSI dataset. The SVM was outperformed by the MLP on the small data, but both had performance degrade after only a few iterations. With the entire data as unlabeled data, the SVM performance improves, outperforming the MLP. The SVM also does not show decrease in accuracy or F over iterations, unlike the MLP, suggesting that EM with SVM is less susceptible to overfitting when given a large amount of unlabeled data. Figure 2 compares F over 30 iterations by co-EM and EM algorithms for both the small and large data sets. Co-EM is generally performing between the two different classifiers in terms of F-measure. There is a benefit in overall accuracy (as shown later), consistent with the EM training objective.

Figure 3 shows the changes of F scores over 30 iterations for the EM and CVEM versions with the two classifiers, again with both small and large unlabeled data sets. As for other approaches, performance degrades after a few iterations when using CVEM on the small data set. On the large data set, CVEM with the SVM performs as well as EM, but improves faster in the earlier iterations. The results for co-EM vs. co-CVEM in Figure 4 shows that with a fixed stopping point neither approach has a clear advantage. Unfortunately, the performance is not stable, so it is not obvious how to determine when to stop otherwise.

Table 1 shows results obtained from the algorithms described in section 3 using a fixed number of iterations (1 on the small data set and 15 for the full set).[2] Results include 3-class accuracy, 3-class average F-measure, and the combined recall of the "positive" (agree, backchannel) and "negative"

[2]On the small data set, overfitting always occurs in 2-3 iterations, but the best stopping point varies on the full data set, with some variants having no loss in 30 iterations. Thus we picked a mid-range stopping point.

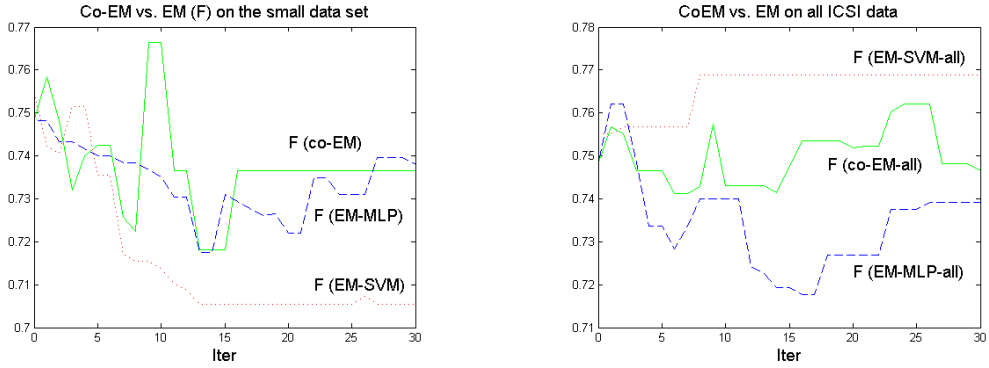

Figure 2: Comparison of F of co-EM with EM-SVM and EM-MLP on the two ICSI data sets

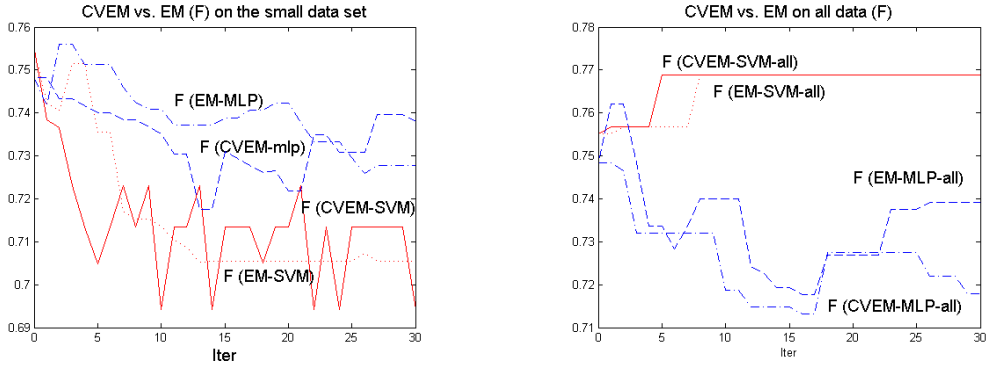

Figure 3: CVEM results (F) the entire ICSI data set.

(disagree) classes (A/D rec), since these classes are most important for conversation analysis. The results are compared to the supervised SVM and MLP models trained with only the labeled subset of the data. In addition, we give results reported for an alternative semi-supervised learning method based on contrast classifiers [**?**], which obtained the best previous results on this data with equivalent features.

On the small data set, the only semi-supervised learning algorithms that gave a gain over the supervised results were co-EM, co-CVEM and the contrast classifier. The two co-EM variants have roughly a 10% improvement in accuracy over the contrast classifier, but part of this gain is due to using a different classifier (the MLP). The contrast classifier has much higher A/D recall. This is likely due to the class imbalance: EM-based and related algorithms prefer majority classes at the expense of reducing the recall of the minority classes. This can be improved by incorporating distribution sensitive learning in these algorithms, as in the contrast classifier. On the large data set, the models EM-SVM and CVEM-SVM give the best F-measure, but the co-EM variants have the best A/D recall.

5 Conclusions

In summary, this work explored several variants of EM and co-training for semi-supervised learning of models for detecting agreement and disagreement in conversational speech. We obtained the best accuracy published so far on the agreement/disagreement classification task by combining MLP and SVM models in co-EM learning. Using cross-validation within the EM iterations (in CVEM and co-CVEM) was investigated in hopes it would help in choosing a stopping point, but it did not yield a single local maximum nor did it have a consistent performance advantage over the standard EM alternatives. Most of the semi-supervised learning algorithms failed to give an improvement

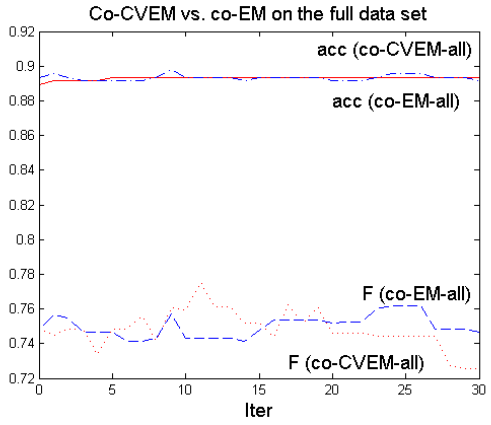

Figure 4: Co-EM vs. Co-CVEM results in acc and F

data set	alg (%)	acc (%)	F rec(%)	A/D
Labeled Set	SVM	88.9	**75.5**	75.3
	MLP	**89.4**	74.8	**79.1**
Small Set	CC	87.1		**82.4**
	EM-SVM	88.9	74.2	75.7
	EM-MLP	89.4	74.8	79.7
	COEM	**89.8**	75.8	80.2
	CVEM-SVM	88.9	73.8	74.7
	CVEM-MLP	89.1	74.2	79.7
	co-CVEM	89.6	**76.2**	79.7
Full Set	EM-SVM-all	89.4	**76.8**	75.8
	EM-MLP-all	88.2	71.9	76.4
	COEM-all	89.4	74.8	**78.0**
	CVEM-SVM-all	89.4	**76.8**	75.8
	CVEM-MLP-all	88.5	71.5	76.9
	co-CVEM-all	**89.8**	75.2	**78.0**

Table 1: Results on the fully labeled subset (supervised training) and on the small and full ICSI data sets (semi-supervised). Best results for each training set and measure are indicated in boldface.

over supervised learning when provided with a small amount of unlabeled data (roughly 5 times the amount of labeled data), with the exception of the co-training variants, and that performance degraded after only a few iterations. With more data (roughly 50 times the amount of labeled data), both the co-EM and SVM variants led to improvements.

The experiments here used only lexical features, so we used two different classifiers rather than different feature sets in co-training. Since we are working with speech data, it is possible that a split of lexical vs. acoustic features would lead to improved performance of co-training, particularly for small data sets. In the task explored here, class distribution skew is an important issue, and detection of infrequent classes can be more effective with controls to account for costs of different errors, as in the contrast classifier.

Acknowledgments

This work is supported in part by the National Science Foundation under grant No. IIS-0326276 and an IBM fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these sponsors.