Topic Learning from Few Examples

Huaiyu Zhu, Shivakumar Vaithyanathan, and Mahesh V. Joshi

IBM Almaden Research Center, 650 Harry Road, San Jose, CA, USA
{huaiyu,vaithyan,joshim}@us.ibm.com

Abstract. This paper describes a semi-supervised algorithm for single class learning with very few examples. The problem is formulated as a hierarchical latent variable model which is clipped to ignore classes not of interest. The model is trained using a multistage EM (msEM) algorithm. The msEM algorithm maximizes the likelihood of the joint distribution of the data and latent variables, under the constraint that the distribution of each layer is fixed in successive stages. We demonstrate that with very few positive examples, the algorithm performs better than training all layers in a single stage. We also show that the latter is equivalent to training a single layer model with corresponding parameters. The performance of the algorithm was verified on several real-world information extraction tasks.

1 Introduction

Several real world problems fall into the category of single class learning, where training data is available for only a single class. Examples of such problems include the identification of a certain class of web-pages - e.g., "personal home pages" or "call for papers" [10]. Building training data for such problems can be a particularly arduous task. A good sample of the positive class must involve all aspects that can lead to inclusion in the positive class. Constructing a negative class would require a uniform representation of the universal set excluding positive class [10].

Information extraction is another area where single class learning problems arise naturally. Information needs of users are too diverse and numerous to allow the creation of significant numbers of labeled examples. Consider, for example, an oil company's corporate reputation management group interested in monitoring articles about its and its competitor's image in the areas of diversity at work place, oil spill issues, environmental policies etc. Obtaining comprehensive examples for every one of these topics is almost impossible. Users are typically willing to provide only very few carefully crafted positive examples for each topic of interest.

The need for single class learning has been recognized and there have been a few previous efforts focusing on learning from positive examples. In [7], the algorithm maps the data using a kernel and then uses the origin as the negative class. In practice this was found to be very sensitive to parametric changes and some heuristic modifications were suggested to include more than just the origin into the negative class [4]. Recently [10] includes unlabeled examples in an iterative framework that identifies examples not sharing features with positive examples, which are then treated as negative examples for training a support vector machine. These approaches have concentrated on identifying negative examples and using them in a discriminative training framework. The motivation

N. Lavrač et al. (Eds.): PKDD 2003, LNAI 2838, pp. 483–494, 2003.

in these approaches has been towards building classifiers that do not degrade in accuracy with the growth in the size of labeled data [10].

Generative modeling approaches have also been applied to the problem of partially labeled data. Unsupervised approaches use joint distributions over the features to identify clusters in the data. In particular, finite mixture models trained using the popular Expectation-Maximization (EM) algorithm [2] have been used extensively. An interesting approach in [5] modifies the EM algorithm to allow the incorporation of labeled data. This approach can, in theory, be used with small amount of labeled data and [5] reported encouraging experiments on multi-class problems where labeled data are available for each class. A variant of this approach to the single class problem, but with larger amounts of labeled data, has been described in [3] with good results.

In this paper we focus on the single-class learning problems with the following two characteristics: (1) The topic of interest only constitutes a very small proportion of candidate data, and (2) The topic is specified by very few positive examples (*seeds*) which usually do not represent a fair sample of the topic. For such problems, single stage clustering algorithms do not perform well: The precision is low unless the number of clusters is large, while the recall is low unless the number of clusters is small. In order to overcome this, we use a hierarchical latent variable model trained with a novel multistage EM (msEM) algorithm. The algorithm concentrates on the class of interest, guided by the labeled examples. Experiments show that the algorithm generalizes well from small number of seeds that form skewed samples of the desired topic.

2 Latent Variable Models and Semi-supervised EM Algorithm

2.1 Latent Variable Model

One commonly used model for clustering is a mixture model of the form

$$p(z) = \sum_a p(z|a) \cdot p(a). \tag{1}$$

where the variable a is a latent variable and is interpreted as class label. Training of this model involves adjusting the parameters of the probability distributions $p(z|a)$ and $p(a)$. This model can be trained effectively using the EM algorithm [2].

Given a dataset $\boldsymbol{z} = \{z_1, z_2, \ldots, z_n\}$ of individual observations of z, the EM algorithm is an iterative algorithm that maximizes the log-likelihood of the model,

$$\sum_i \log p(z_i) = \sum_i \log \sum_a p(z_i, a). \tag{2}$$

2.2 Semi-supervised EM Algorithm

The EM algorithm for maximizing (2) is an *unsupervised* algorithm. However, some *prior* information is available. E.g., in single class classification, we often have some *seed* information – a few labeled examples for the class of interest. Incorporating such *seed constraints* into the EM algorithm results in a *semi-supervised EM algorithm* (ssEM) [5]. We use the version as shown in Fig. 1.

– Set initial model parameters for the distributions $p(z|a)$ and $p(a)$.
– Iterate until convergence over the following two steps:
E-Step: For each z_i, compute $q(a|z_i)$ by Bayes rule and seed constraint

$$\begin{cases} q(a|z_i) = p(a|z_i) = p(z_i, a)/p(z_i), & z_i \notin \text{Seeds} \\ q(a=1|z_i) = 1, & z_i \in \text{Seeds}. \end{cases} \tag{3}$$

M-Step: Estimate new parameters for $p(z|a)$ and $p(a)$ by maximizing

$$\sum_{i,a} q(a|z_i) \log p(z_i, a). \tag{4}$$

Fig. 1. Semi-supervised EM algorithm (ssEM)

However, with very few labeled examples, seed constraints alone are not sufficient to tackle the problem at hand. For the task of identifying a single class from multiple possibilities, there is a trade-off between the number of components in the mixture model (1) and the precision of the chosen class. If the number of components in the mixture model is small, the chosen component is likely to contain a large number of spurious datapoints. If the number of components is large, the desired class might be fragmented among many different components. We now proceed to describe more powerful models and algorithms to address this.

2.3 Hierarchical Latent Variable Models

A two level hierarchical model can be obtained by replacing the model (1) with

$$p(z) = \sum_{a_0, a_1} p(z|a_0, a_1) \cdot p(a_1|a_0) \cdot p(a_0), \tag{5}$$

where a_0 and a_1 are two levels of latent variables in the hierarchy.

A full-blown model of the form (5) can be expensive to train due to the combinatorial effect of the hierarchical hidden variables in the E-step. Since we are only interested in a single class, it is intuitively plausible that clipping off the branches in the hierarchical model not corresponding to the class of interest would reduce a substantial amount of computation without large impact on performance. This suggests using the following "clipped model" (assuming $a = 1$ corresponds to the class of interest)

$$p(z_i) = \sum_{a_0=1} p(a_0) \sum_{a_1} p(a_1|a_0) \cdot p(z_i|a_0, a_1) + \sum_{a_0 \neq 1} p(a_0) p(z_i|a_0). \tag{6}$$

However, as the following two lemmas show, training either of them with ssEM does not exhibit advantage over single layer models. The full potential of the clipped model can be realized by a new training algorithm, which we propose in Section 3.

Lemma 1. The hierarchical model (5) can be represented as a single layer model. They both behave identically under standard EM training algorithm.

Proof. The model (5) is a marginal distribution of

$$p(z_i, a_0, a_1) = p(z_i|a_0, a_1)p(a_0, a_1) \tag{7}$$

Suppose the combination (a_0, a_1) takes n distinct values. For example, if a_0 takes n_0 values and a_1 takes n_1 values, then $n = n_0 n_1$. Introduce a variable c with n distinct values, the distribution (5) is identical to

$$p(z_i) = \sum_c p(z_i|c)p(c). \tag{8}$$

The derivation of the training algorithm involves expressions of $p(z_i|a_0, a_1)$ and $p(a_0, a_1)$, which can be replaced with expressions of $p(z_i|c)$ and $p(c)$. The resulting training algorithm is therefore equivalent after simple renaming of parameters. □

Lemma 2. The clipped hierarchical model (6) can be represented as a single layer model. They both behave identically under standard EM training algorithm.

Proof. Since a_1 does not occur for $a_0 \neq 1$, we can arbitrarily set $a_1 = 1$ for $a_0 \neq 1$. Then (6) is a marginal of

$$p(z_i, a_0, a_1) = p(z_i|a_0, a_1)p(a_0, a_1) \tag{9}$$

where $p(z_i|a_0, a_1) = p(z_i|a_0)$ and $p(a_0, a_1) = p(a_0)$ for $a_1 = 1$. This case is analogous to that of Lemma 1, except that (a_0, a_1) would take value in a discrete set with $2n - 1$ elements. Hereafter the proof follows that of Lemma 1. □

3 Training Clipped Model with Multistage EM Algorithm

For problems involving large number of components in the mixture model, the semi-supervised EM can be successfully applied when labeled data are available for different classes [5]. However, a more powerful algorithm is needed when only a few labeled datapoints are available for just one class. We propose a multistage EM algorithm (msEM) to train the clipped model, which is more suitable for such problems.

3.1 Generalized Form of Likelihood

The log-likelihood (2) can be written in a more general form [1]

$$N \sum_i q(z_i) \log \sum_a p(z_i, a), \tag{10}$$

where $q(z_i) = 1/N$ is the *empirical distribution* of the data and N is the size (number of datapoints) of the dataset. Using (10), the M-step (4) of the EM algorithm can now be written as maximizing

$$\sum_{i,a} q(z_i)q(a|z_i) \log p(z_i, a). \tag{11}$$

The convergence properties of the EM algorithm still hold even when $q(z_i)$ is not the uniform distribution over the observed data $\boldsymbol{z}$ [1]. The EM algorithm can therefore be regarded as a mapping $q(z) \to q(z, a)$.

- For each layer $m = 0, 1, \ldots$, set initial model parameters for $p_m(z|a)$ and $p_m(a)$.
- Set the first layer empirical distribution $q_0(z_i) = q(z_i) = 1/N$ for all datapoint z_i.
- Iterate until convergence:
  - For each layer $m = 0, 1, \ldots$, carry out the following three steps
    - **E-Step:** For each z_i, compute $q_m(a_m|z_i)$ by Bayes rule and seed constraint

$$\begin{cases} q_m(a_m|z_i) = p_m(a_m|z_i) = p_m(z_i, a_m)/p_m(z_i), & z_i \notin \text{Seeds} \\ q_m(a_m = 1|z_i) = 1, & z_i \in \text{Seeds}. \end{cases} \tag{12}$$

    - **M-Step:** Estimate new parameters for $p_m(z|a)$ and $p_m(a)$ by maximizing

$$\sum_{i,a_m} q_m(z_i) q_m(a_m|z_i) \log p_m(z_i, a_m). \tag{13}$$

    - Set empirical distribution for the next layer $q_{m+1}(z_i) = q_m(z_i|a_m = 1)$ for all datapoint z_i using Bayes rule

$$q_m(z_i|a_m) = q_m(z_i) q_m(a_m|z_i)/q_m(a_m). \tag{14}$$

Fig. 2. Multistage semi-supervised EM algorithm (msEM)

3.2 Multistage Semi-supervised EM Algorithm

The msEM algorithm for the clipped model trains each layer successively by incorporating the empirical distribution from the previous layers (Fig. 2).

Comparing the E and M-steps of the above algorithm, (12) and (13), with those of the ssEM algorithm, (3) and (11), it can be seen that the computation for layer m implements the EM algorithm that maximizes the generalized log-likelihood

$$\sum_i q_m(z_i) \log \sum_{a_m} p_m(z_i, a_m), \tag{15}$$

where $q_m(z_i)$ is given by $q_m(z_i|a_m = 1)$. The intuition behind this algorithm is that, by weighting each datapoint with $q_m(z_i)$ instead of the uniform distribution, we are deemphasizing those z_i that are less likely to be in class 1 as predicted by layer $m - 1$. The discrimination in layer m could then conceivably concentrate more on the finer details not addressed at layer $m - 1$. Each layer acts as a regularizer to restrict the variability of the model in the next layer.

3.3 Deriving the Updated Empirical Distribution

In the msEM algorithm, the empirical distribution q_m of layer m is computed from the results of layer $m - 1$. The rule for computing q_m can be derived from a global optimization problem involving layers 0 through m. The objective function for layer m is

$$\sum_{i,a_0,\ldots,a_{m-1}} q(z_i, a_0, \ldots, a_{m-1}) \log \sum_{a_m} p(z_i, a_0, \ldots, a_m). \tag{16}$$

Maximizing (16) for successive m with ssEM implies that layer m is trained under the constraint that $q(z_i, a_0, \ldots, a_{m-1})$ is fixed. We now show that this is indeed equivalent to the msEM given in Fig. 2.

For layer 0, the objective function (16) reduces to

$$\sum_i q(z_i) \log \sum_{a_0} p(z_i, a_0). \tag{17}$$

The ssEM algorithm for maximizing (17) is the same as the msEM algorithm for layer 0, with the following substitutions,

$$q(z_i, a_0) = q_0(z_i, a_0), \qquad p(z_i, a_0) = p_0(z_i, a_0). \tag{18}$$

For layer 1, the objective function (16) reduces to

$$\sum_{i, a_0} q(z_i, a_0) \log \sum_{a_1} p(z_i, a_0, a_1). \tag{19}$$

The E- and M-steps corresponding to (3) and (11) are therefore

$$\begin{cases} q(a_1|a_0, z_i) = p(a_1|a_0, z_i) = p(z_i, a_1|a_0)/p(z_i|a_0), & z_i \notin \text{Seeds} \\ q(a_1 = 1|a_0, z_i) = 1, & z_i \in \text{Seeds}. \end{cases} \tag{20}$$

$$\text{maximize} \sum_{z_i, a_0, a_1} q(z_i, a_0) q(a_1|z_i, a_0) \log p(z_i, a_0, a_1). \tag{21}$$

Since the model is clipped, it is clear that the E-step (20) is equivalent to (12) with the following substitutions,

$$p_1(a_1, z_i) = p(a_1, z_i|a_0 = 1), \qquad q_1(a_1, z_i) = q(a_1, z_i|a_0 = 1). \tag{22}$$

The M-step objective function in (21) can be expanded into three terms,

$$\begin{aligned} &\sum_{a_0} q(a_0) \log p(a_0) + \sum_{a_0 \neq 1} \sum_{z_i} q(z_i, a_0) \log p(z_i|a_0) \\ &+ \sum_{a_0 = 1} \sum_{z_i, a_1} q(a_0) q(z_i|a_0) q(a_1|z_i, a_0) \log p(z_i, a_1|a_0). \end{aligned} \tag{23}$$

Maximizing the first two terms leads to $p(z_i, a_0)$ already calculated in (18). Using the definitions of p_1 and q_1 in (22) and the fact that the third term only involves $a_0 = 1$, it can be verified that maximizing the third term is equivalent to maximizing (13), provided that the empirical distribution is given as $q_1(z_i) = q_0(z_i|a_0 = 1)$. Therefore we have proved the equivalence for layer 1.

Similarly it can be shown that for any m, the E- and M-steps in the msEM are equivalent to the steps in an ssEM that maximizes (16), provided that $q_{m+1}(z_i) = q_m(z_i|a_m = 1)$.

3.4 Considerations on the Convergence of Multistage EM

In our experiments the msEM always converged with speed similar to ssEM. Here we outline an approach to prove the convergence. It is known that each iteration of the standard EM algorithm increases the likelihood, which converges to a stationary point [2]. Under certain conditions, this also results in the convergence of the probability distributions p and q to fixed points [8]. Under certain stronger conditions, the mapping $q(z) \to q(z, a)$ is continuous. Assuming that each layer of the clipped model satisfies all these conditions, the convergence of msEM can be proved by induction. Layer 0 implements ssEM so $q_0(z, a)$ converges. Suppose $q_{m-1}(z, a)$ converges. Then $q_m(z) = q_{m-1}(z|a = 1)$ also converges. The continuity of the mapping $q_m(z) \to q_m(z, a)$ then implies the convergence of $q_m(z, a)$.

3.5 Multistage EM Interpreted as Reverse Boosting

We note briefly the relationship between the msEM algorithm and boosted density estimation [6]. An extensive description of this relationship is available from [11].

Let $a_{0:m} = (a_0, \ldots, a_m)$ and denote by $a_{0:m} = 1$ the condition $a_0 = \cdots = a_{m-1} = 1$. The msEM algorithm can be regarded as building successively more complex models with weighted weak learners. At stage m the model built so far is

$$\begin{aligned} P_m(z) = & \sum_{a_0 \neq 1} p(a_0)p(z|a_0) \\ & + p(a_0 = 1) \sum_{a_1 \neq 1} p(a_1|a_0 = 1)p(z|a_0 = 1, a_1) \\ & + \ldots \\ & + \prod_{l=0}^{m-1} p(a_l = 1|a_{0:l-1} = 1) \sum_{a_m} p(a_m|a_{0:m-1} = 1)p(z|a_{0:m-1} = 1, a_m). \end{aligned} \tag{24}$$

The msEM algorithm attempts to improve the classification of a single class by getting successively better density estimates for that single class subject to the seed constraints. The weak learner chosen at layer m is a finite mixture model $p_m(z, a_m)$, trained with ssEM. In contrast to the boosted density estimation [6], which emphasizes regions with

1. Set the initial weights $w_i = 1/N$.
2. for m=1 to M
   (a) Use EM algorithm to compute $p_m(z_i|a_m)$ and $p_m(a_m)$ that maximizes $\sum_i w_i \log \sum_a p_m(z_i|a_m)p_m(a_m)$ subject to *seed constraints*, obtaining $q_m(a_m|z_i)$ in the process.
   (b) Set $w_i = q_m(z_i|a_m = 1)$, calculated with Bayes rule (14).
3. Output final model P_M

Fig. 3. Multistage EM interpreted as reverse boosting

Table 1. Experimental datasets and topics

Dataset	# Entities	# Datapoints	# Features	Half-window	Topics	# Seeds
TN	2151	87,251	24,808	400 characters	"chip", "web"	3
OP	10	30,000	77,762	75 words	"diversity"	2–8

Intel has not reduced its capital spending budget of $7.5 billion for the year, in part to accommodate the introduction of 300-millimeter wafer production. Chips produced on the new wafers will also be made with the more advanced 0.13-micron manufacturing process and contain copper wires. **Intel** *currently makes its chips with the 0.18-micron manufacturing process and uses aluminum. The micron measurements refer to the size of features on the chip. The shift will result in smaller, cooler, faster and cheaper processors. "Intel expects chips produced on 300-millimeter wafers to cost 30 percent less than those made using the smaller wafers, " Tom Garrett, Intel's 300-millimeter program manager, said in a statement.*

Fig. 4. Example passage for the chip manufacturing topic

high uncertainty, the msEM emphasizes regions with high certainty of being in the class of interest, by increasing the weight of datapoints that perform well in the previous layer. The msEM algorithm in boosting framework is shown in Fig. 3, where the weight w_i corresponds to $q_{m+1}(z_i)$ in the msEM algorithm.

4 Experimental Setting

4.1 Data Sets

We experimented with msEM and several existing algorithms on two real-world document collections. The datasets are formed from passages in documents crawled from the Web. For each document, a set of proper names are identified as being of interest (named entities). A passage of a fixed window size surrounding each named entity is taken as its context. An example of such a passage is shown in Fig. 4. The context is tokenized into words (discarding punctuations), removing stop words (a list of 232 common words) and stemming (using Porter's Stemmer), resulting in a vector of feature counts. The named entity and the context feature counts together form a datapoint z_i. Note that each document can provide multiple datapoints.

Two test collections were made (Table 1). The collection TN was gathered from the Tech News section of CNet (`www.cnet.com`). A named entity tagger was used to identify organizational names (such as Intel, IBM, Microsoft etc.). The collection OP consists of web-pages discussing oil companies. A list of organizational names (each containing variations of the same organizational name) was obtained from industry experts. Some of these datasets will soon be made publicly available for other researchers to test their algorithms.

4.2 Topics and Seeds

The experiments were conducted on the following three topics:

chip Steps taken by semiconductor manufacturers to produce cheaper, faster and thermally more efficient microprocessors and microchips. (Dataset TN)
web Web Service protocols for business process integration. (Dataset TN).
diversity Issues related to diversity at work-place. (Dataset OP).

For the "chip" topic, three seeds occurring in two documents were identified from the corpus as relevant to the query. For the "web" topic, three seeds from three documents were selected as seeds. The "diversity" topic was used extensively to understand the behavior of the different algorithms. A series of experiments with several seed sets of different characteristics were used, as described below.

4.3 Algorithms, Parameters, and Evaluations

As comparison to msEM, the results are evaluated against several alternative algorithms: semi-supervised EM algorithm on single-layer latent variable models (ssEM)[5], a simple nearest neighbor algorithm (NN)[9] and a proximity pattern search (PPS).

For the ssEM algorithm, we used $p(z|a) = p(x|a)p(y|a)$ where x and y are the named entity and the context feature count vector, respectively, $p(x|a)$ is a discrete distribution, and $p(y|a)$ is the multinomial distribution. Laplace's smoothing is used in the M-step [5]. Let k be the numbers of components in the mixture model. The parameters of $p(z|a)$ are initialized by assigning $q(a = j|z_i) = 1/k + \epsilon$, $j = 1, \ldots, k$, for non-seed z_i and $q(a = 1|z_i) = 1$ for seed z_i, followed by an M-step, where ϵ is a small noise used for breaking symmetry among the components.

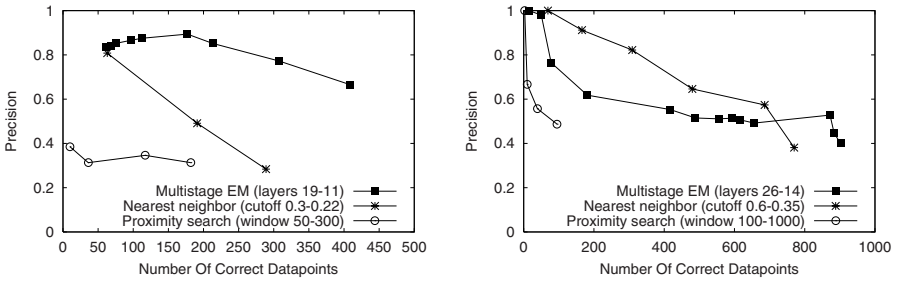
Each layer in the msEM algorithm uses the same model as in the ssEM algorithm with exactly two components ($k = 2$). The initializing of each layer is also the same as above, except that no symmetry breaking is necessary.

The NN algorithm is performed on the same tokenized context (without the named entity) using cosine similarity on the feature count vectors. A datapoint is deemed on topic if the similarity is higher than a cutoff value.

For the topics "chip" and "web" we also tested a type of proximity pattern search (PPS), based on patterns given by domain experts. A data point is deemed on topic if the pattern matches within a distance of the named entity in the original (not tokenized) context.

Each of these algorithms contains a parameter that controls the trade-off between precision and recall: number of layers for msEM, number of nodes for ssEM, similarity cutoff for NN and the distance for PPS. Results for different values of the control parameters are shown in the next section.

Results of the algorithms were manually evaluated by domain experts to establish a set of ground truth. Since the datasets are too large to be evaluated completely, only results from the high-precision versions of each algorithm are pooled together and evaluated. This allows us to calculate the precision of these algorithms and the number of correctly retrieved datapoints, but not the actual recall (which requires knowing the total number of on-topic datapoints in the whole corpus).



(a) Topic "chip" (b) Topic "web"

Fig. 5. Precision vs number of correct datapoints

5 Results

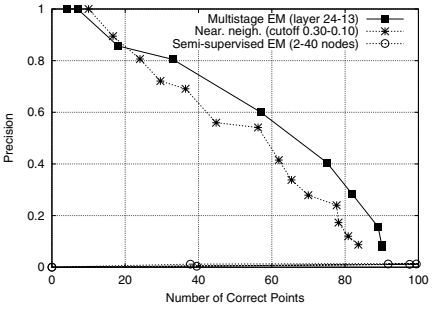
Our experiments with ssEM [5] consistently returns very poor results. The algorithm was run for several different number of components ranging from 20 to 100 and the results were significantly worse than those of the other algorithms. We therefore do not report the actual numbers in this paper.

5.1 Results on the TN Collection

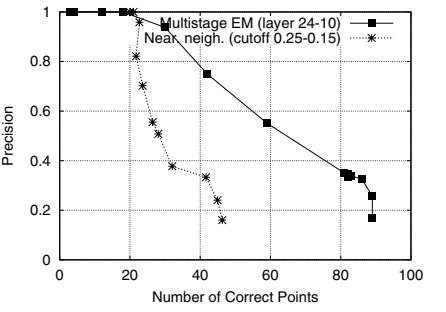
The results obtained by the algorithms on different parameter settings on the TN dataset are shown in 5(a) and 5(b). An immediate glance of these figures indicates that msEM is a clear winner for topic "chip", outperforming the other algorithms significantly. For topic "web" the NN algorithm is very competitive and the results drop off at 0.35. A look at the results of the pattern-matching proximity search helps shed more light on the relative performance between the msEM and nearest neighbor. For topic "chip" the best performance for proximity search is a precision of 0.4 which drops to a low of 0.3 with increasing recall – while for topic "web" the best performance is about 0.9 dropping to a low of just below 0.45. This seems to suggest that topic "web" is defined by simpler patterns than topic "chip". Closer examination of the results and discussion with the domain expert revealed that the "web" topic was particularly narrow that can be effectively defined by spotting a few keywords. In effect the smoothing/generalization effect provided by the msEM algorithm did not provide any advantages – instead worsened the results.

5.2 Effects of Seeds

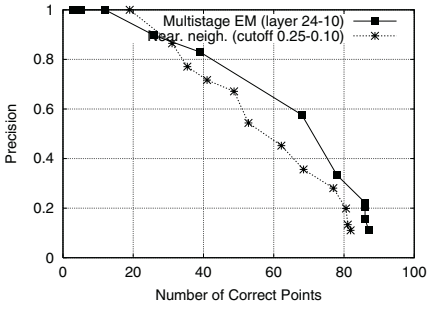
To better understand the effect of seeds on the two algorithms msEM and NN, we conducted several experiments on the "diversity" topic using the dataset OP. This topic consists of several subtopics such as issues concerning minority, domestic partner benefit, gender equality, etc. The complexity of the task is increased due to the fact that these subtopics share very few words in common. We identified two types of seeds:



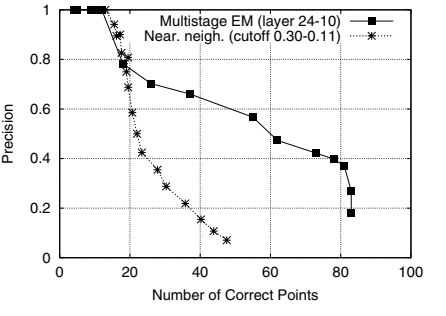
(a) Two seeds, both general



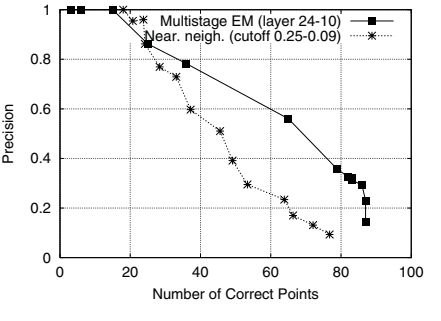
(b) Three seeds, one general, two specific



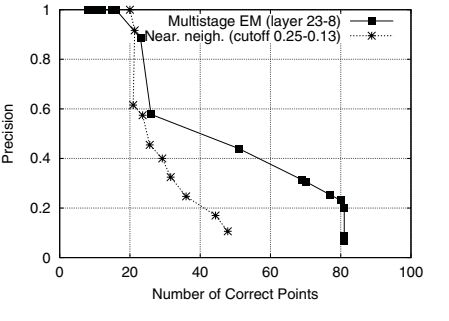
(c) Three seeds, two general, one specific



(d) Three seeds, all specific



(e) Two seeds, one general, one specific



(f) Eight specific seeds

Fig. 6. Precision vs number of correct datapoints for Topic "diversity"

– General seeds: passages discussing work place diversity policies in general.

– Specific seeds: passages discussing a specific instance of a company changing its policy on domestic partner benefits.

The results of NN and msEM are shown in Fig. 6. Several observations can be made. The NN is almost always better at highest precision. The msEM is almost always better at generalization. With general seeds, both NN and msEM perform comparably. With specific seeds, the NN algorithm almost exclusively retrieves datapoints that deal with the same specific instance as the seeds (confirmed by examining the actual retrieved passages). In contrast, at intermediate precision, the msEM can generalize significantly better than NN. The better performance of the NN at the range of very high precision and low recall is due to its retrieving only datapoints very similar to the seeds. At this range the generalization ability of the msEM is not particularly useful. On the hand, the NN fails to generalize for the specific seeds, which forms a skewed sample of the topic. The msEM is able to generalize better because its retrieval set is not entirely defined by similarity to seeds — the clustering of the unlabeled data also plays an important role.

Acknowledgments

We would like to thank Sreeram Balakrishnan, Christopher Campbell, Ashutosh Garg, Jussi Myllymaki and Wayne Niblack for their help in various stages of this work.

References

1. S.I. Amari. Information geometry of the EM and em algorithms for neural networks. *Neural Networks*, 8(9):1379–1408, 1995.
2. A. P. Dempster, N. Laird, and D. Rubin. Maximum likelihood for incomplete data via the EM algorithm. *J. of the Royal Statistical Society, ser. B*, 39:1–38, 1977.
3. Bing Liu, Wee Sun Lee, Philip Yu, and Xiaoli Li. Partially supervised classification of text documents. In *International Conference On Machine Learning*, 2002.
4. Larry Manevitz and Malik Yousef. One class SVMs for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.
5. Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.
6. Saharon Rosset and Eran Segal. Boosting density estimation. In *NIPS*, 2002.
7. B. Scholkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1472, 2001.
8. C. Wu. On the convergence properties of the EM algorithm. *Annals of Statistics*, 11:95–103, 1983.
9. Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, 1999.
10. Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. PEBL: Positive example based learning for web page classification using SVM. In *Proceedings of 2002 SIGKDD Conference*, pages 239–248, 2002.
11. Huaiyu Zhu and Shivakumar Vaithyanathan. A multistage EM algorithm for training reduced hierarchical latent variable models. Technical Report RJ-10283, IBM Research Report, January 2003.