Homotopy-based Semi-Supervised Hidden Markov Models for Sequence Labeling*

Gholamreza Haffari and **Anoop Sarkar**
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
{ghaffar1,anoop}@cs.sfu.ca

Abstract

This paper explores the use of the homotopy method for training a semi-supervised Hidden Markov Model (HMM) used for sequence labeling. We provide a novel polynomial-time algorithm to trace the local maximum of the likelihood function for HMMs from full weight on the labeled data to full weight on the unlabeled data. We present an experimental analysis of different techniques for choosing the best balance between labeled and unlabeled data based on the characteristics observed along this path. Furthermore, experimental results on the field segmentation task in information extraction show that the Homotopy-based method significantly outperforms EM-based semi-supervised learning, and provides a more accurate alternative to the use of held-out data to pick the best balance for combining labeled and unlabeled data.

1 Introduction

In semi-supervised learning, given a sample containing both labeled data L and unlabeled data U, the maximum likelihood estimator Θ^{mle} maximizes:

$$\mathcal{L}(\Theta) := \sum_{(\mathbf{x},\mathbf{y})\in L} \log P(\mathbf{x},\mathbf{y}|\Theta) + \sum_{\mathbf{x}\in U} \log P(\mathbf{x}|\Theta) \quad (1)$$

where $\mathbf{y}$ is a structured output label, e.g. a sequence of tags in the part-of-speech tagging task, or parse trees in the statistical parsing task. When the number of labeled instances is very small compared to the unlabeled instances, i.e. $|L| \ll |U|$, the likelihood of labeled data is dominated by that of unlabeled data, and the valuable information in the labeled data is almost completely ignored.

Several studies in the natural language processing (NLP) literature have shown that as the size of unlabeled data increases, the performance of the model with Θ^{mle} may deteriorate, most notably in (Merialdo, 1993; Nigam et al., 2000). One strategy commonly used to alleviate this problem is to explicitly weigh the contribution of labeled and unlabeled data in (1) by $\lambda \in [0, 1]$. This new parameter controls the influence of unlabeled data but is estimated either by (a) an ad-hoc setting, where labeled data is given more weight than unlabeled data, or (b) by using the EM algorithm or (c) by using a held-out set. But each of these alternatives is problematic: the ad-hoc strategy does not work well in general; the EM algorithm ignores the labeled data almost entirely; and using held-out data involves finding a good step size for the search, but small changes in λ may cause drastic changes in the estimated parameters and the performance of the resulting model. Moreover, if labeled data is scarce, which is usually the case, using a held-out set wastes a valuable resource[1].

In this paper, we use continuation techniques (Corduneanu and Jaakkola, 2002) for determining λ for structured prediction tasks involving HMMs, and more broadly, the product of multinomials (PoM) model. We provide a polynomial-time algorithm for HMMs to trace the local maxima of the likelihood function from full weight on the labeled data to full weight on the unlabeled data. In doing so, we introduce dynamic programming algorithms for HMMs that enable the efficient computation over unlabeled data of the covariance between pairs of state transition counts and pairs of state-state and state-observation counts. We present a detailed experimental analysis of different techniques for choosing the best balance be-

* We would like to thank Shihao Ji and the anonymous reviewers for their comments. This research was supported in part by NSERC, Canada.

[1] Apart from these reasons, we also provide an experimental comparision between the homotopy based approach, the EM algorithm, and the use of a held out set.

tween labeled and unlabeled data based on the characteristics observed along this path. Furthermore, experimental results on the field segmentation task in information extraction show that the Homotopy-based method significantly outperforms EM-based semi-supervised learning, and provides a more accurate alternative to the use of held-out data to pick the best balance for combining labeled and unlabeled data. We argue this approach is a *best bet* method which is robust to different settings and types of labeled and unlabeled data combinations.

2 Homotopy Continuation

A continuation method embeds a given hard root finding problem $G(\Theta) = 0$ into a family of problems $H(\Theta(\lambda), \lambda) = 0$ parameterized by λ such that $H(\Theta(1), 1) = 0$ is the original given problem, and $H(\Theta(0), 0) = 0$ is an easy problem $F(\Theta) = 0$ (Richter and DeCarlo, 1983). We start from a solution Θ_0 for $F(\Theta) = 0$, and deform it to a solution Θ_1 for $G(\Theta) = 0$ while keeping track of the solutions of the intermediate problems[2]. A simple deformation or homotopy function is:

$$H(\Theta, \lambda) = (1 - \lambda)F(\Theta) + \lambda G(\Theta) \qquad (2)$$

There are many ways to define a homotopy map, but it is not trivial to always guarantee the existence of a path of solutions for the intermediate problems. Fortunately for the homotopy map we will consider in this paper, the path of solutions which starts from $\lambda = 0$ to $\lambda = 1$ exists and is unique.

In order to find the path numerically, we seek a curve $\Theta(\lambda)$ which satisfies $H(\Theta(\lambda), \lambda) = 0$. This is found by differentiating with respect to λ and solving the resulting differential equation. To handle singularities along the path and to be able to follow the path beyond them, we introduce a new variable s (which in our case is the unit path length) and solve the following differential equation for $(\Theta(s), \lambda(s))$:

$$\frac{\partial H(\Theta, \lambda)}{\partial \Theta}\frac{d\Theta}{ds} + \frac{\partial H(\Theta, \lambda)}{\partial \lambda}\frac{d\lambda}{ds} = 0 \qquad (3)$$

subject to $||(\frac{d\Theta}{ds}, \frac{d\lambda}{ds})||_2 = 1$ and the initial condition $(\Theta(0), \lambda(0)) = (\Theta_0, 0)$. We use the Euler method (see Algorithm 1) to solve (3) but higher order methods such as Runge-Kutta of order 2 or 3 can also be used.

[2]This deformation gives us a solution *path* $(\Theta(\lambda), \lambda)$ in $\mathbb{R}^{d+1}$ for $\lambda \in [0, 1]$, where each component of the d-dimensional solution vector $\Theta(\lambda) = (\theta_1(\lambda), .., \theta_d(\lambda))$ is a function of λ.

3 Homotopy-based Parameter Estimation

One way to control the contribution of the labeled and unlabeled data is to parameterize the log likelihood function as $\mathcal{L}_\lambda(\Theta)$ defined by

$$\frac{1-\lambda}{|L|}\sum_{(x,y)\in L}\log P(x, y|\Theta) + \frac{\lambda}{|U|}\sum_{x\in U}\log P(x|\Theta)$$

How do we choose the best λ? An operator called $EM_\lambda$is used with the property that its fixed points (locally) maximize $\mathcal{L}_\lambda(\Theta)$. Starting from a fixed point of EM_λ when λ is zero[3], the path of fixed point of this operator is *followed* for $\lambda > 0$ by continuation techniques. Finally the best value for λ is chosen based on the characteristics observed along the path. One option is to choose an allocation value where the first critical[4] point occurs had we followed the path based on λ, i.e. without introducing s (see Sec. 2). Beyond the first critical point, the fixed points may not have their roots in the starting point which has all the information from labeled data (Corduneanu and Jaakkola, 2002). Alternatively, an allocation may be chosen which corresponds to the model that gives the maximum entropy for label distributions of unlabeled instances (Ji et al., 2007). In our experiments, we compare all of these methods for determining the choice of λ.

3.1 Product of Multinomials Model

Product of Multinomials (PoM) model is an important class of probabilistic models especially for NLP which includes HMMs and PCFGs among others (Collins, 2005). In the PoM model, the probability of a pair $(\mathbf{x}, \mathbf{y})$ is

$$P(\mathbf{x}, \mathbf{y}|\Theta) = \prod_{m=1}^{M}\prod_{\omega\in\Omega_m}\Theta_m(\omega)^{Count(\mathbf{x},\mathbf{y},\omega)} \qquad (4)$$

where $Count(\mathbf{x}, \mathbf{y}, \omega)$ shows how many times an outcome $\omega \in \Omega_m$ has been seen in the input-output pair $(\mathbf{x}, \mathbf{y})$, and M is the total number of multinomials. A multinomial distribution parameterized

[3]In general, EM_0 can have multiple local maxima, but in our case, EM_0 has only one global maximum, found analytically using relative frequency estimation.

[4]A critical point is where a discontinuity or bifurcation occurs. In our setting, almost all of the critical points correspond to discontinuities (Corduneanu, 2002).

by Θ_m is put on each discrete space Ω_m where the probability of an outcome ω is denoted by $\Theta_m(\omega)$. So for each space Ω_m, we have $\sum_{\omega\in\Omega_m}\Theta_m(\omega) = 1$.

Consider an HMM with K states. There are three types of parameters: (i) initial state probabilities $P(s)$ which is a multinomial over states $\Theta_0(s)$, (ii) state transition probabilities $P(s'|s)$ which are K multinomials over states $\Theta_s(s')$, and (iii) emission probabilities $P(a|s)$ which are K multinomials over observation alphabet $\Theta_{s+K}(a)$. To compute the probability of a pair $(\mathbf{x},\mathbf{y})$, normally we go through the sequence and multiply the probability of the seen state-state and state-observation events:

$$P(\mathbf{x},\mathbf{y}|\Theta) = \Theta_0(y_0)\Theta_{y_1+K}(x_1)\prod_{t=2}^{|y|}\Theta_{y_{t-1}}(y_t)\Theta_{y_t+K}(x_t)$$

which is in the form of (4) if it is written in terms of the multinomials involved.

3.2 EM_λ Operator for the PoM Model

Usually EM is used to maximize $\mathcal{L}(\Theta)$ and estimate the model parameters in the situation where some parts of the training data are hidden. EM has an intuitive description for the PoM model: starting from an arbitrary value for parameters, iteratively update the probability mass of each event proportional to its count in labeled data plus its *expected* count in the unlabeled data, until convergence.

By changing the EM's update rule, we get an algorithm for maximizing $\mathcal{L}_\lambda(\Theta)$:

$$\tilde{\Theta}_m(\omega) = \frac{1-\lambda}{|L|}\sum_{(\mathbf{x},\mathbf{y})\in L} Count(\mathbf{x},\mathbf{y},\omega) + \frac{\lambda}{|U|}\sum_{\mathbf{x}\in U}\sum_{\mathbf{y}\in\mathcal{Y}_\mathbf{x}} Count(\mathbf{x},\mathbf{y},\omega)P(\mathbf{y}|\mathbf{x},\Theta^{old}) \quad (5)$$

where $\tilde{\Theta}_m$ is the unnormalized parameter vector, i.e. $\Theta_m(\omega) = \frac{\tilde{\Theta}_m(\omega)}{\sum_{\omega\in\Omega_m}\tilde{\Theta}_m(\omega)}$. The expected counts can be computed efficiently based on the forward-backward recurrence for HMMs (Rabiner, 1989) and inside-outside recurrence for PCFGs (Lari and Young, 1990). The right hand side of (5) is an operator we call EM_λ which transforms the old parameter values to their new (unnormalized) values. EM_0 and EM_1 correspond respectively to purely supervised and unsupervised parameter estimation settings, and:

$$EM_\lambda(\Theta) = (1-\lambda)EM_0(\Theta) + \lambda EM_1(\Theta) \quad (6)$$

3.3 Homotopy for the PoM Model

The iterative maximization algorithm, described in the previous section, proceeds until it reaches a fixed point $EM_\lambda(\Theta) = \tilde{\Theta}$, where based on (6):

$$(1-\lambda)\underbrace{(\tilde{\Theta} - EM_0(\Theta))}_{F(\Theta)} + \lambda\underbrace{(\tilde{\Theta} - EM_1(\Theta))}_{G(\Theta)} = 0 \quad (7)$$

The above condition governs the (local) maxima of EM_λ. Comparing to (2) we can see that (7) can be viewed as a homotopy map.

We can generalize (7) by replacing $(1-\lambda)$ with a function $g_1(\lambda)$ and λ with $g_2(\lambda)$[5]. This corresponds to other ways of balancing labeled and unlabeled data log-likelihoods in (1). Moreover, we may partition the parameter set and use the homotopy method to just estimate the parameters in one partition while keeping the rest of parameters fixed (to inject some domain knowledge to the estimation procedure), or repeat it through partitions. We will see this in Sec. 5.2 where the transition matrix of an HMM is frozen and the emission probabilities are learned with the continuation method.

Algorithm 1 describes how to use continuation techniques used for homotopy maps in order to trace the path of fixed points for the EM_λ operator. The algorithm uses the Euler method to solve the following differential equation governing the fixed points of EM_λ:

$$\begin{bmatrix} \lambda\nabla_{\tilde{\Theta}}EM_1(\Theta) - I & EM_1(\Theta) - EM_0 \end{bmatrix}\begin{bmatrix} d\tilde{\Theta} \\ d\lambda \end{bmatrix} = 0$$

For PoM models $\nabla_{\tilde{\Theta}}EM_1(\Theta)$ can be written compactly as follows[6]:

$$\frac{1}{|U|}\sum_{x\in U} COV_{P(\mathbf{y}|\mathbf{x},\Theta)}\Big[Count(\mathbf{x},\mathbf{y})\Big]\cdot\mathbf{H} \quad (8)$$

where $COV_{P(\mathbf{y}|\mathbf{x},\Theta)}[Count(\mathbf{x},\mathbf{y})]$ is the conditional covariance matrix of *all* features $Count(\mathbf{x},\mathbf{y},\omega)$ given an unlabeled instance $\mathbf{x}$. We denote the entry corresponding to events ω_1 and ω_2 of this matrix by $COV_{P(\mathbf{y}|\mathbf{x},\Theta)}(\omega_1,\omega_2)$; $\mathbf{H}$ is a block diagonal matrix built from $\mathbf{H}_{\Omega_i}$ where

$$\mathbf{H}_{\Omega_i} = (\tilde{\Theta}_i(\alpha_1),..,\tilde{\Theta}_i(\alpha_{|\Omega_i|}))\cdot\mathbf{I} - \frac{\mathbf{1}_{|\Omega_i|\times|\Omega_i|}}{\sum_{\alpha\in\Omega_i}\tilde{\Theta}_i(\alpha)}$$

[5]However the following two conditions must be satisfied: (i) the deformation map is reduced to $(\tilde{\Theta} - EM_0(\Theta))$ at $\lambda = 0$ and $(\tilde{\Theta} - EM_1(\Theta))$ at $\lambda = 1$, and (ii) the path of solutions exists for Eqn. (2).

[6]A full derivation is provided in (Haffari and Sarkar, 2008)

Algorithm 1 Homotopy Continuation for EM_λ

Input: Labeled data set L
Input: Unlabeled data set U
Input: Step size δ
Initialize $[\tilde{\Theta} \quad \lambda] = [EM_0 \quad 0]$ based on L
$\eta^{old} \leftarrow [\mathbf{0} \quad 1]$
repeat
  Compute $\nabla_{\tilde{\Theta}} EM_1(\Theta)$ and $EM_1(\Theta)$ based on unlabeled data U
  Compute $\eta = [d\tilde{\Theta} \quad d\lambda]$ as the kernel of $[\lambda \nabla_{\tilde{\Theta}} EM_1(\Theta) - I \quad EM_1(\Theta) - EM_0]$
  if $\eta \cdot \eta^{old} < 0$ **then**
    $\eta \leftarrow -\eta$
  end if
  $[\tilde{\Theta} \quad \lambda] \leftarrow [\tilde{\Theta} \quad \lambda] + \delta \frac{\eta}{||\eta||_2}$
  $\eta^{old} \leftarrow \eta$
until $\lambda \geq 1$

Computing the covariance matrix in (8) is a challenging problem because it consists of summing quantities over all possible structures $\mathcal{Y}_\mathbf{x}$ associated with each unlabeled instance $\mathbf{x}$, which is exponential in the size of the input for HMMs.

4 Efficient Computation of the Covariance Matrix

The entry $COV_{P(\mathbf{y}|\mathbf{x},\Theta)}(\omega_1, \omega_2)$ of the features covariance matrix is

$$\mathbf{E}[Count(\mathbf{x}, \mathbf{y}, \omega_1) Count(\mathbf{x}, \mathbf{y}, \omega_2)] - \mathbf{E}[Count(\mathbf{x}, \mathbf{y}, \omega_1)] \mathbf{E}[Count(\mathbf{x}, \mathbf{y}, \omega_2)]$$

where the expectations are taken under $P(\mathbf{y}|\mathbf{x}, \Theta)$. To efficiently calculate the covariance, we need to be able to efficiently compute the expectations. The *linear* count expectations can be computed efficiently by the forward-backward recurrence for HMMs. However, we have to design new algorithms for *quadratic* count expectations which will be done in the rest of this section.

We add a special **begin** symbol to the sequences and replace the initial probabilities with $P(s|\textbf{begin})$. Based on the terminology used in (4), the outcomes belong to two categories: $\omega = (s, s')$ where state s' follows state s, and $\omega = (s, a)$ where symbol a is emitted from state s. Define the feature function $f_\omega(\mathbf{x}, \mathbf{y}, t)$ to be 1 if the outcome ω happens at time step t, and 0 otherwise. Based on the fact that $Count(\mathbf{x}, \mathbf{y}, \omega) = \sum_{t=1}^{|\mathbf{x}|} f_\omega(\mathbf{x}, \mathbf{y}, t)$, we have

$$\mathbf{E}[Count(\mathbf{x}, \mathbf{y}, \omega_1) Count(\mathbf{x}, \mathbf{y}, \omega_2)] = \sum_{t_1} \sum_{t_2} \sum_{\mathbf{y} \in \mathcal{Y}_x} f_{\omega_1}(\mathbf{x}, \mathbf{y}, t_1) f_{\omega_2}(\mathbf{x}, \mathbf{y}, t_2) P(\mathbf{y}|\mathbf{x}, \Theta)$$

which is the summation of $|\mathbf{x}|^2$ different expectations. Fixing two positions t_1 and t_2, each expectation is the probability (over all possible labels) of observing ω_1 and ω_2 at these two positions respectively, which can be efficiently computed using the following data structure. Prepare an auxiliary table $Z^\mathbf{x}$ containing $P(x_{[i+1,j]}, s_i, s_j)$, for every pair of states s_i and s_j for all positions i, j ($i \leq j$):

$$Z^\mathbf{x}_{i,j}(s_i, s_j) = \sum_{s_{i+1}, .., s_{j-1}} \prod_{k=i}^{j-1} P(s_{k+1}|s_k) P(x_{k+1}|s_{k+1})$$

Let matrix $M^\mathbf{x}_k = [M^\mathbf{x}_k(s, s')]$ where $M^\mathbf{x}_k(s, s') = P(s'|s) P(x_k|s')$; then $Z^\mathbf{x}_{i,j} = \prod_{k=i}^{j-1} M^\mathbf{x}_k$. Forward and backward probabilities can also be computed from $Z^\mathbf{x}$, so building this table helps to compute both linear and quadratic count expectations.

With this table, computing the quadratic counts is straightforward. When both events are of type state-observation, i.e. $\omega = (s, a)$ and $\omega' = (s', a')$, their expected quadratic count can be computed as

$$\sum_{t_1} \sum_{t_2} \delta_{x_{t_1}, a} \delta_{x_{t_2}, a'} \Big[\sum_k P(k|\textbf{begin}) Z^\mathbf{x}_{1,t_1}(k, s) . Z^\mathbf{x}_{t_1,t_2}(s, s') . \sum_k Z^\mathbf{x}_{t_2,n}(s', k) P(\textbf{end}|k) \Big]$$

where $\delta_{x_t,a}$ is 1 if x_t is equal to a and 0 otherwise. Likewise we can compute the expected quadratic counts for other combination of events: (i) both are of type state-state, (ii) one is of type state-state and the other state-observation.

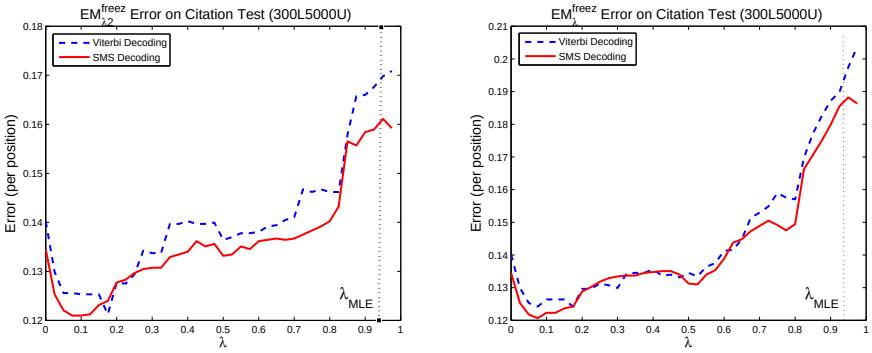
There are $\frac{L(L+1)}{2}$ tables needed for a sequence of length L, and the time complexity of building each of them is $\mathcal{O}(K^3)$ where K is the number of states in the HMM. When computing the covariance matrix, the observations are fixed and there is no need to consider all possible combinations of observations and states. The most expensive part of the computation is the situation where the two events are of type state-state which amounts to $\mathcal{O}(L^2 K^4)$ matrix updates. Noting that a single entry needs $\mathcal{O}(K)$ for its updating, the time complexity of computing expected quadratic counts for a single sequence is $\mathcal{O}(L^2 K^5)$. The space needed to store the auxiliary tables is $\mathcal{O}(L^2 K^2)$ and the space needed for covariance matrix is $\mathcal{O}((K^2 + NK)^2)$ where N is the alphabet size.

5 Experimental Results

In the field segmentation task, a document is considered to be a sequence of fields. The goal is

[*EDITOR* A. Elmagarmid, editor.] [*TITLE* Transaction Models for Advanced Database Applications] [*PUBLISHER* Morgan-Kaufmann,] [*DATE* 1992.]

Figure 1: A field segmentation example for Citations dataset.



(a) (b)

Figure 2: EM_λ error rates while increasing the allocation from 0 to 1 by the step size 0.025.

to segment the document into fields, and to label each field. In our experiments we use the bibliographic citation dataset described in (Peng and McCallum, 2004) (see Fig. 1 for an example of the input and expected label output for this task). This dataset has 500 annotated citations with 13 fields; 5000 unannotated citations were added to it later by (Grenager et al., 2005). The annotated data is split into a 300-document training set, a 100-document development (dev) set, and a 100-document test set[7].

We use a first order HMM with the size of hidden states equal to the number of fields (equal to 13). We freeze the transition probabilities to what has been observed in the labeled data and only learn the emission probabilities. The transition probabilities are kept frozen due to the nature of this task in which the transition information can be learned with very little labeled data, e.g. first start with 'author' then move to 'title' and so on. However, the challenging aspect of this dataset is to find the segment spans for each field, which depends on learning the emission probabilities, based on the fixed transition probabilities.

At test time, we use both Viterbi (most probable sequence of states) decoding and sequence of most probable states decoding methods, and abbreviate them by Viterbi and SMS respectively. We report results in terms of precision, recall and F-measure for finding the citation fields, as well as accuracy calculated per position, i.e. the ratio of the words labeled correctly for sequences to all of the words. The segment-based precision and recall scores are, of course, lower than the accuracy computed on the per-token basis. However, both these numbers need to be taken into account in order to understand performance in the field segmentation task. Each input word sequence in this task is very long (with an average length of 36.7) but the number of fields to be recovered is a small number comparatively (on average there are 5.4 field segments in a sentence where the average length of a segment is 6.8). Even a few one-word mistakes in finding the full segment span leads to a drastic fall in precision and recall. The situation is quite different from part-of-speech tagging, or even noun-phrase chunking using sequence learning methods. Thus, for this task both the per-token accuracy as well as the segment precision and recall are equally important in gauging performance.

Smoothing to remove zero components in the starting point is crucial otherwise these features do not generalize well and yet we know that they have been observed in the unlabeled data. We use a simple add-ϵ smoothing, where ϵ is .2 for transition table entries and .05 for the emission table entries. In all experiments, we deal with unknown words in test data by replacing words seen less than 5 times in training by the unknown word token.

5.1 Problems with MLE

MLE chooses to set $\lambda = \frac{|U|}{|L|+|U|}$ which almost ignores labeled data information and puts all the weight on the unlabeled data[8]. To see this empirically, we show the per position error rates at dif-

[7] From http://www.stanford.edu/grenager/data/unsupie.tgz

[8] One anonymous reviewer suggests using $\lambda = \frac{|L|}{|L|+|U|}$ but the "best bet" for different tasks that we mention in the Introduction may not necessarily be a small λ value.

ferent source allocation for HMMs trained on 300 labeled and 5000 unlabeled sequences for the Citation dataset in Fig. 2(a). For each allocation we have run EM_λ algorithm, initialized to smoothed counts from labeled data, until convergence. As the plots show, initially the error decreases as λ increases; however, it starts to increase after λ passes a certain value. MLE has higher error rates compared to complete data estimate, and its performance is far from the best way of combining labeled and unlabeled data.

In Fig. 2(b), we have done similar experiment with the difference that for each value of λ, the starting point of the EM_λ is the final solution found in the previous value of λ. As seen in the plot, the intermediate local optima have better performance compared to the previous experiment, but still the imbalance between labeled and unlabeled data negatively affects the quality of the solutions compared to the purely supervised solution.

The likelihood surface is non-convex and has many local maxima. Here EM performs hill climbing on the likelihood surface, and arguably the resulting (locally optimal) model may not reflect the quality of the globally optimal MLE. But we conjecture that even the MLE model(s) which globally maximize the likelihood may suffer from the problem of the size imbalance between labeled and unlabeled data, since what matters is the influence of unlabeled data on the likelihood. (Chang et. al., 2007) also report on using hard-EM on these datasets[9] in which the performance degrades compared to the purely supervised model.

5.2 Choosing λ in Homotopy-based HMM

We analyze different criteria in picking the best value of λ based on inspection of the continuation path. The following criteria are considered:

• **monotone:** The first iteration in which the monotonicity of the path is changed, or equivalently the first iteration in which the determinant of $\lambda \nabla_{\tilde{\Theta}} EM_1(\Theta) - I$ in Algorithm 1 becomes zero (Corduneanu and Jaakkola, 2002).

• **minEig:** Instead of looking into the determinant of the above matrix, consider its minimum eigenvalue. Across all iterations, choose the one for which this minimum eigenvalue is the lowest.

• **maxEnt:** Choose the iteration whose model puts the maximum entropy on the labeling distribution for unlabeled data (Ji et al., 2007).

[9]In Hard-EM, the probability mass is fully assigned to the most probable label, instead of all possible labels.

The second criterion is new, and experimentally has shown a good performance; it indicates the amount of singularity of a matrix.

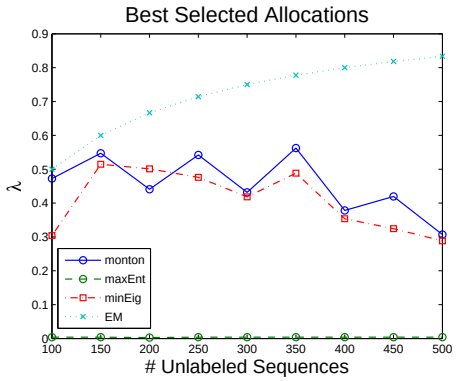


Figure 3: λ values picked by different methods. The size of the labeled data is fixed to 100, and results are averaged over 4 runs. The λ values picked by MaxEnt method for 500 unlabeled examples was .008.

We fix 100 labeled sequences and vary the number of unlabeled sequences from 100 to 500 by a step of 50. All of the experiments are repeated four times with different randomly chosen unlabeled datasets, and the results are the average over four runs. The chosen allocations based on the described criteria are plotted in Figure 3, and their associated performance measures can be seen in Figure 4.

Figure 3 shows that as the unlabeled data set grows, the reliance of 'minEig' and 'monotone' methods on unlabeled data decreases whereas in EM it increases. The 'minEig' method is more conservative than 'monotone' in that it usually chooses smaller λ values. The plots in Figure 4 show that homotopy-based HMM always outperforms EM-based HMM. Moreover, 'maxEnt' method outperforms other ways of picking λ. However, as the size of the unlabeled data increases, the three methods tend to have similar performances.

5.3 Homotopy v.s. other methods

In the second set of experiments, we compare the performance of the homotopy based method against the competitive methods for picking the value of λ.

We use all of the labeled sequences (size is 300) and vary the number of unlabeled sequences from 300 to 1000 by the step size of 100. For the first competitive method, 100 labeled sequences are put in a held out set and used to select the best value

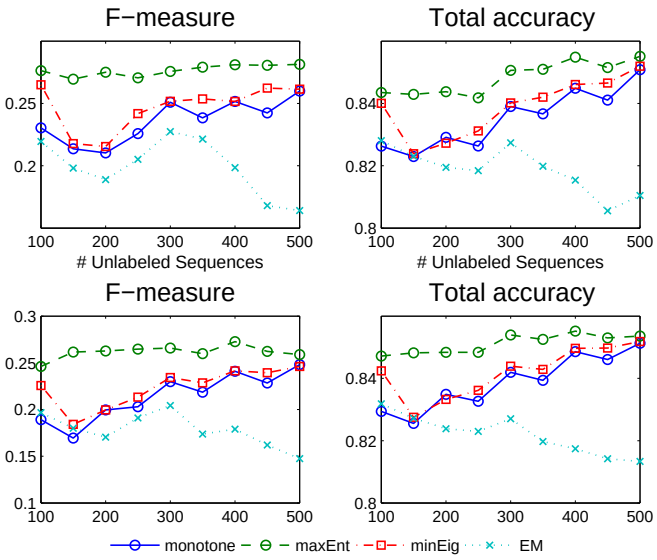


Figure 4: The comparison of different techniques for choosing the best allocation based on datasets with 100 labeled sequences and varying number of unlabeled sequences. Each figure shows the average over 4 runs. F-measure is calculated based on the *segments*, and total accuracy is calculated based on tokens in individual positions. The two plots in the top represent Viterbi decoding, and the two plots in the bottom represent SMS decoding.

of λ based on brute-force search using a fixed step size; afterwards, this value is used to train HMM (based on 200 remaining labeled sequences and unlabeled data). The second competitive method, which we call 'Oracle', is similar to the previous method except we use the test set as the held out set and all of the 300 labeled sequences as the training set. In a sense, the resulting model is the best we can expect from cross validation based on the knowledge of true labels for the test set. Despite the name 'Oracle', in this setting the λ value is selected based on the log-likelihood criterion, so it is possible that the 'Oracle' method is outperformed by another method in terms of precision/recall/f-score. Finally, EM is considered as the third baseline.

The results are summarized in Table 1. When decoding based on SMS, the homotopy-based HMM outperforms the 'Held-out' method for all of performance measures, and generally behaves better than the 'Oracle' method. When decoding based on Viterbi, the accuracy of the homotopy-based HMM is better than 'Held-out' and is in the same range as the 'Oracle'; the three methods have roughly the same f-score. The λ value found by Homotopy gives a small weight to unlabeled data, and so it might seem that it is ignoring the unlabeled data. This is not the case, even with a small weight the unlabeled data has an impact, as can be seen in the comparison with the purely Supervised baseline in Table 1 where the Homotopy method outperforms the Supervised baseline by more than 3.5 points of f-score with SMS-decoding. Homotopy-based HMM with SMS-decoding outperforms all of the other methods.

We noticed that accuracy was better for 700 unlabeled examples in this dataset, and so we include those results as well in Table 1. We observed some noise in unlabeled sequences; so as the size of the unlabeled data set grows, this noise increases as well. In addition to finding the right balance between labeled and unlabeled data, this is another factor in semi-supervised learning. For each particular unlabeled dataset size (we experimented using 300 to 1000 unlabeled data with a step size of 100) the Homotopy method outperforms the other alternatives.

6 Related Previous Work

Homotopy based parameter estimation was originally proposed in (Corduneanu and Jaakkola, 2002) for Naïve Bayes models and mixture of Gaussians, and (Ji et al., 2007) used it for HMM-based sequence *classification* which means that an input sequence $\mathbf{x}$ is classified into a class label $y \in \{1, \ldots, k\}$ (the class label is not structured, i.e. not a sequence of tags). The classification is done using a collection of k HMMs by computing $\Pr(\mathbf{x}, y \mid \Theta^y)$ which sums over all states in each HMM Θ^y for input $\mathbf{x}$. The algorithms in (Ji et al., 2007) could be adapted to the task of sequence labeling, but we argue that our algorithms provide a straightforward and direct solution.

There have been some studies using the Citation dataset, but it is not easy to directly compare their results due to differences in preprocessing, the amount of the previous knowledge and rich features used by the models, and the training data which were used. (Chang et. al., 2007) used a first order HMM in order to investigate injecting prior domain knowledge to self-training style bootstrapping by encoding human knowledge into declarative constraints. (Grenager et al., 2005) used a first order HMM which has a diagonal transition matrix and a specialized boundary model. In both works, the number of *randomly* selected labeled and unlabeled training data is varied, which makes a di-

	size of unlab data	λ	Viterbi decoding		SMS decoding	
			p, r, f-score	accuracy	p, r, f-score	accuracy
Homotopy	700	.004	.292, .290, .290	87.1%	.321, .332, **.326**	**89%**
	1000	.004	.292, .291, .291	87.9%	.296, .298, .296	88.6%
Held-out	700	.220	.311, .291, .297	87.1%	.295, .288, .289	87.2%
	1000	.320	.300, .276, .283	86.9%	.308, .281, .287	87.2%
Oracle	700	.150	.284, .293, .287	87.8%	.295, .313, .303	88%
	1000	.200	.285, .294, .289	87.9%	.277, .292, .284	88.7%
EM	700	.700	.213, .211, .211	84.8%	.213, .220, .216	85.2%
	1000	.770	.199, .198, .198	83.7%	.187, .198, .192	83.6%
Supervised	0	0	.281, .278, .279	87%	.298, .280, .288	88.4%

Table 1: Results using entire labeled data with segment precision/recall/f-score and token based accuracy.

rect numerical comparison impossible. (Peng and McCallum, 2004) used only labeled data to train conditional random fields and HMMs with second order state transitions where they allow observation in each position to depend on the current state as well as observation of the previous position.

7 Conclusion

In many NLP tasks, the addition of unlabeled data to labeled data can *decrease* the performance on that task. This is often because the unlabeled data can overwhelm the information obtained from the labeled data. In this paper, we have described a methodology and provided efficient algorithms for an approach that attempts to ensure that unlabeled data does not hurt performance. The experimental results show that homotopy-based training performs better than other commonly used competitive methods. We plan to explore faster ways for computing the (approximate) covariance matrix, e.g., label sequences can be sampled from $P(\mathbf{y}|\mathbf{x}, \Theta)$ and an approximation of the covariance matrix can be computed based on these samples. Also, it is possible to compute the covariance matrix in polynomial-time for labels which have richer interdependencies such as those generated by a context free grammars (Haffari and Sarkar, 2008). Finally, in Algorithm 1 we used a fixed step size; the number of iterations in the homotopy path following can be reduced greatly with adaptive step size methods (Allgower and Georg, 1993).

References

E. L. Allgower, K. Georg 1993. Continuation and Path Following, *Acta Numerica*, 2:1-64.

M. Chang and L. Ratinov and D. Roth. 2007. Guiding Semi-Supervision with Constraint-Driven Learning, ACL 2007.

M. Collins 2005. Notes on the EM Algorithm, NLP course notes, MIT.

A. Corduneanu. 2002. Stable Mixing of Complete and Incomplete Information, Masters Thesis, MIT.

A. Corduneanu and T. Jaakkola. 2002. Continuation Methods for Mixing Heterogeneous Sources, UAI 2002.

T. Grenager, D. Klein, and C. Manning. 2005. Unsupervised Learning of Field Segmentation Models for Information Extraction, ACL 2005.

G. Haffari and A. Sarkar. 2008. A Continuation Method for Semi-supervised Learning in Product of Multinomials Models, Technical Report. Simon Fraser University. School of Computing Science.

K. Lari, and S. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm, *Computer Speech and Language* (4).

S. Ji, L. Watson and L. Carin. 2007. Semi-Supervised Learning of Hidden Markov Models via a Homotopy Method, manuscript.

B. Merialdo. 1993. Tagging English text with a probabilistic model, *Computational Linguistics*

K. Nigam, A. McCallum, S. Thrun and T. Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, 39. p. 103-134.

F. Peng and A. McCallum. 2004. Accurate Information Extraction from Research Papers using Conditional Random Fields, HLT-NAACL 2004.

L. Rabiner. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proc. of the IEEE*, 77(2).

S. Richter, and R. DeCarlo. 1983. Continuation methods: Theory and applications, *IEEE Trans. on Automatic Control*, Vol 26, issue 6.