Why doesn't EM find good HMM POS-taggers?

Mark Johnson

Microsoft Research
Redmond, WA
t-majoh@microsoft.com

Brown University
Providence, RI
Mark_Johnson@Brown.edu

Abstract

This paper investigates why the HMMs estimated by Expectation-Maximization (EM) produce such poor results as Part-of-Speech (POS) taggers. We find that the HMMs estimated by EM generally assign a roughly equal number of word tokens to each hidden state, while the empirical distribution of tokens to POS tags is highly skewed. This motivates a Bayesian approach using a sparse prior to bias the estimator toward such a skewed distribution. We investigate Gibbs Sampling (GS) and Variational Bayes (VB) estimators and show that VB converges faster than GS for this task and that VB significantly improves 1-to-1 tagging accuracy over EM. We also show that EM does nearly as well as VB when the number of hidden HMM states is dramatically reduced. We also point out the high variance in all of these estimators, and that they require many more iterations to approach convergence than usually thought.

1 Introduction

It is well known that Expectation-Maximization (EM) performs poorly in unsupervised induction of linguistic structure (Carroll and Charniak, 1992; Merialdo, 1994; Klein, 2005; Smith, 2006). In retrospect one can certainly find reasons to explain this failure: after all, likelihood does not appear in the wide variety of linguistic tests proposed for identifying linguistic structure (Fromkin, 2001).

This paper focuses on unsupervised part-of-speech (POS) tagging, because it is perhaps the simplest linguistic induction task. We suggest that one reason for the apparent failure of EM for POS tagging is that it tends to assign relatively equal numbers of tokens to each hidden state, while the empirical distribution of POS tags is highly skewed, like many linguistic (and non-linguistic) phenomena (Mitzenmacher, 2003). We focus on first-order Hidden Markov Models (HMMs) in which the hidden state is interpreted as a POS tag, also known as bitag models.

In this setting we show that EM performs poorly when evaluated using a "1-to-1 accuracy" evaluation, where each POS tag corresponds to at most one hidden state, but is more competitive when evaluated using a "many-to-1 accuracy" evaluation, where several hidden states may correspond to the same POS tag. We explain this by observing that the distribution of hidden states to words proposed by the EM-estimated HMMs is relatively uniform, while the empirical distribution of POS tags is heavily skewed towards a few high-frequency tags. Based on this, we propose a Bayesian prior that biases the system toward more skewed distributions and show that this raises the 1-to-1 accuracy significantly. Finally, we show that a similar increase in accuracy can be achieved by reducing the number of hidden states in the models estimated by EM.

There is certainly much useful information that bitag HMMs models cannot capture. Toutanova et al. (2003) describe a wide variety of morphological and distributional features useful for POS tagging, and Clark (2003) proposes ways of incorporating some of these in an unsupervised tagging model. However, bitag models are rich enough to capture at least some distributional information (i.e., the tag

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 296–305, Prague, June 2007. ©2007 Association for Computational Linguistics

for a word depends on the tags assigned to its neighbours). Moreover, more complex models add additional complicating factors that interact in ways still poorly understood; for example, smoothing is generally regarded as essential for higher-order HMMs, yet it is not clear how to integrate smoothing into unsupervised estimation procedures (Goodman, 2001; Wang and Schuurmans, 2005).

Most previous work exploiting unsupervised training data for inferring POS tagging models has focused on semi-supervised methods in the in which the learner is provided with a lexicon specifying the possible tags for each word (Merialdo, 1994; Smith and Eisner, 2005; Goldwater and Griffiths, 2007) or a small number of "prototypes" for each POS (Haghighi and Klein, 2006). In the context of semi-supervised learning using a tag lexicon, Wang and Schuurmans (2005) observe discrepencies between the empirical and estimated tag frequencies similar to those observed here, and show that constraining the estimation procedure to preserve the empirical frequencies improves tagging accuracy. (This approach cannot be used in an unsupervised setting since the empirical tag distribution is not available). However, as Banko and Moore (2004) point out, the accuracy achieved by these unsupervised methods depends strongly on the precise nature of the supervised training data (in their case, the ambiguity of the tag lexicon available to the system), which makes it more difficult to understand the behaviour of such systems.

2 Evaluation

All of the experiments described below have the same basic structure: an estimator is used to infer a bitag HMM from the unsupervised training corpus (the words of Penn Treebank (PTB) Wall Street Journal corpus (Marcus et al., 1993)), and then the resulting model is used to label each word of that corpus with one of the HMM's hidden states. This section describes how we evaluate how well these sequences of hidden states correspond to the gold-standard POS tags for the training corpus (here, the PTB POS tags). The chief difficulty is determining the correspondence between the hidden states and the gold-standard POS tags.

Perhaps the most straightforward method of establishing this correspondence is to deterministically map each hidden state to the POS tag it co-occurs most frequently with, and return the proportion of the resulting POS tags that are the same as the POS tags of the gold-standard corpus. We call this the *many-to-1 accuracy* of the hidden state sequence because several hidden states may map to the same POS tag (and some POS tags may not be mapped to by any hidden states at all).

As Clark (2003) points out, many-to-1 accuracy has several defects. If a system is permitted to posit an unbounded number of hidden states (which is not the case here) then it can achieve a perfect many-to-1 accuracy by placing every word token into its own unique state. Cross-validation, i.e., identifying the many-to-1 mapping and evaluating on different subsets of the data, would answer many of these objections. Haghighi and Klein (2006) propose constraining the mapping from hidden states to POS tags so that at most one hidden state maps to any POS tag. This mapping is found by greedily assigning hidden states to POS tags until either the hidden states or POS tags are exhausted (note that if the number of hidden states and POS tags differ, some will be unassigned). We call the accuracy of the POS sequence obtained using this map its *1-to-1 accuracy*.

Finally, several authors have proposed using information-theoretic measures of the divergence between the hidden state and POS tag sequences. Goldwater and Griffiths (2007) propose using the *Variation of Information* (VI) metric described by Meilă (2003). We regard the assignments of hidden states and POS tags to the words of the corpus as two different ways of clustering those words, and evaluate the conditional entropy of each clustering conditioned on the other. The VI is the sum of these conditional entropies. Specifically, given a corpus labeled with hidden states and POS tags, if $\tilde{p}(y), \tilde{p}(t)$ and $\tilde{p}(y,t)$ are the empirical probabilities of a hidden state y, a POS tag t, and the cooccurance of y and t respectively, then the mutual information I, entropies H and variation of information VI are defined as follows:

$$\begin{aligned}
H(Y) &= -\sum_{y} \tilde{p}(y) \log \tilde{p}(y) \\
H(T) &= -\sum_{t} \tilde{p}(t) \log \tilde{p}(t) \\
I(Y,T) &= \sum_{y,t} \tilde{p}(y,t) \log \frac{\tilde{p}(y,t)}{\tilde{p}(y)\tilde{p}(t)} \\
H(Y|T) &= H(Y) - I(Y,T)
\end{aligned}$$

$$
\begin{aligned}
H(T|Y) &= H(T) - I(Y,T) \\
VI(Y,T) &= H(Y|T) + H(T|Y)
\end{aligned}
$$

As Meilă (2003) shows, VI is a metric on the space of probability distributions whose value reflects the divergence between the two distributions, and only takes the value zero when the two distributions are identical.

3 Maximum Likelihood via Expectation-Maximization

There are several excellent textbook presentations of Hidden Markov Models and the Forward-Backward algorithm for Expectation-Maximization (Jelinek, 1997; Manning and Schütze, 1999; Bishop, 2006), so we do not cover them in detail here. Conceptually, a Hidden Markov Model generates a sequence of observations $\mathbf{x} = (x_0, \ldots, x_n)$ (here, the words of the corpus) by first using a Markov model to generate a sequence of hidden states $\mathbf{y} = (y_0, \ldots, y_n)$ (which will be mapped to POS tags during evaluation as described above) and then generating each word x_i conditioned on its corresponding state y_i. We insert endmarkers at the beginning and ending of the corpus and between sentence boundaries, and constrain the estimator to associate endmarkers with a state that never appears with any other observation type (this means each sentence can be processed independently by first-order HMMs; these endmarkers are ignored during evaluation).

In more detail, the HMM is specified by multinomials θ_y and ϕ_y for each hidden state y, where θ_y specifies the distribution over states following y and ϕ_y specifies the distribution over observations x given state y.

$$
\begin{array}{llll}
y_i & | \; y_{i-1} = y & \sim & \mathrm{Multi}(\theta_y) \\
x_i & | \; y_i = y & \sim & \mathrm{Multi}(\phi_y)
\end{array}
\qquad (1)
$$

We used the Forward-Backward algorithm to perform Expectation-Maximization, which is a procedure that iteratively re-estimates the model parameters (θ, ϕ), converging on a local maximum of the likelihood. Specifically, if the parameter estimate at time ℓ is $(\theta^{(\ell)}, \phi^{(\ell)})$, then the re-estimated parameters at time $\ell + 1$ are:

$$
\begin{aligned}
\theta^{(\ell+1)}_{y'|y} &= \mathrm{E}[n_{y',y}]/\mathrm{E}[n_y] \qquad (2) \\
\phi^{(\ell+1)}_{x|y} &= \mathrm{E}[n_{x,y}]/\mathrm{E}[n_y]
\end{aligned}
$$

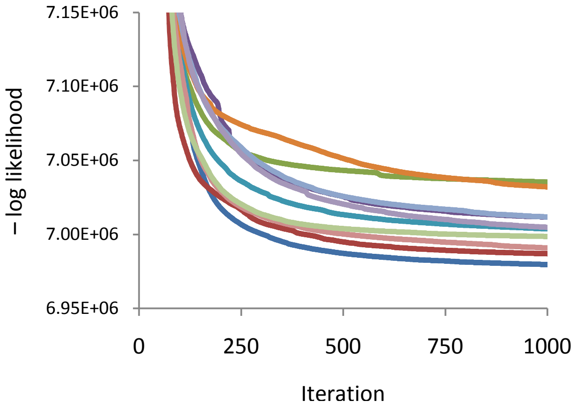

Figure 1: Variation in negative log likelihood with increasing iterations for 10 EM runs from different random starting points.

where $n_{x,y}$ is the number of times observation x occurs with state y, $n_{y',y}$ is the number of times state y' follows y and n_y is the number of occurences of state y; all expectations are taken with respect to the model $(\theta^{(\ell)}, \phi^{(\ell)})$.

We took care to implement this and the other algorithms used in this paper efficiently, since optimal performance was often only achieved after several hundred iterations. It is well-known that EM often takes a large number of iterations to converge in likelihood, and we found this here too, as shown in Figure 1. As that figure makes clear, likelihood is still increasing after several hundred iterations.

Perhaps more surprisingly, we often found dramatic changes in accuracy in the order of 5% occuring after several hundred iterations, so we ran 1,000 iterations of EM in all of the experiments described here; each run took approximately 2.5 days computation on a 3.6GHz Pentium 4. It's well-known that accuracy often decreases after the first few EM iterations (which we also observed); however in our experiments we found that performance improves again after 100 iterations and continues improving roughly monotonically. Figure 2 shows how 1-to-1 accuracy varies with iteration during 10 runs from different random starting points. Note that 1-to-1 accuracy at termination ranges from 0.38 to 0.45; a spread of 0.07.

We obtained a dramatic speedup by working directly with probabilities and rescaling after each observation to avoid underflow, rather than working with log probabilities (thanks to Yoshimasa Tsu-

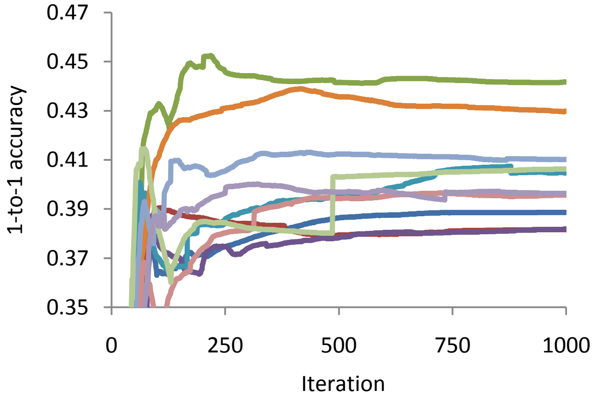

Figure 2: Variation in 1-to-1 accuracy with increasing iterations for 10 EM runs from different random starting points.

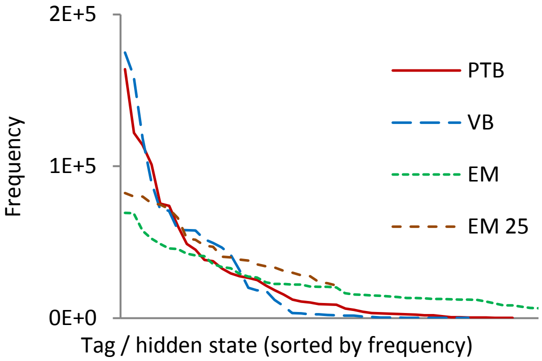

Figure 3: The average number of words labeled with each hidden state or tag for the EM, VB (with $\alpha_x = \alpha_y = 0.1$) and EM-25 estimators (EM-25 is the EM estimator with 25 hidden states).

ruoka for pointing this out). Since we evaluated the accuracy of the estimated tags after each iteration, it was important that decoding be done efficiently as well. While most researchers use Viterbi decoding to find the most likely state sequence, maximum marginal decoding (which labels the observation x_i with the state y_i that maximizes the marginal probability $\mathrm{P}(y_i|\mathbf{x}, \theta, \phi)$) is faster because it re-uses the forward and backward tables already constructed by the Forward-Backward algorithm. Moreover, in separate experiments we found that the maximum marginal state sequence almost always scored higher than the Viterbi state sequence in all of our evaluations, and at modest numbers of iterations (up to 50) often scored more than 5% better.

We also noticed a wide variance in the performance of models due to random initialization (both θ and ϕ are initially jittered to break symmetry); this wide variance was observed with all of the estimators investigated in this paper. This means we cannot compare estimators on the basis of single runs, so we ran each estimator 10 times from different random starting points and report both mean and standard deviation for all scores.

Finally, we also experimented with annealing, in which the parameters θ and ϕ are raised to the power $1/T$, where T is a "temperature" parameter that is slowly lowered toward 1 at each iteration according to some "annealing schedule". We experimented with a variety of starting temperatures and annealing schedules (e.g., linear, exponential, etc), but were unable to find any that produced models whose likelihoods were significantly higher (i.e., the models fit better) than those found without annealing.

The evaluation of the models produced by the EM and other estimators is presented in Table 1. It is difficult to compare these with previous work, but Haghighi and Klein (2006) report that in a completely unsupervised setting, their MRF model, which uses a large set of additional features and a more complex estimation procedure, achieves an average 1-to-1 accuracy of 41.3%. Because they provide no information about the variance in this accuracy it is difficult to tell whether there is a significant difference between their estimator and the EM estimator, but it is clear that when EM is run long enough, the performance of even very simple models like the bitag HMM is better than generally recognized.

As Table 1 makes clear, the EM estimator produces models that are extremely competitive in many-to-1 accuracy and Variation of Information, but are significantly worse in 1-to-1 accuracy. We can understand these results by comparing the distribution of words to hidden states to the distribution of words to POS tags in the gold-standard evaluation corpus. As Figure 3 shows, the distribution of words to POS tags is highly skewed, with just 6 POS tags, NN, IN, NNP, DT, JJ and NNS, accounting for over 55% of the tokens in the corpus. By contrast, the EM distribution is much flatter. This also explains why the many-to-1 accuracy is so much better than the one-to-one accuracy; presumably several hidden

Estimator		1-to-1		Many-to-1		*VI*		$H(T\|Y)$		$H(Y\|T)$	
EM	(50)	0.40	(0.02)	**0.62**	(0.01)	4.46	(0.08)	**1.75**	(0.04)	2.71	(0.06)
VB$(0.1, 0.1)$	(50)	**0.47**	(0.02)	0.50	(0.02)	4.28	(0.09)	2.39	(0.07)	1.89	(0.06)
VB$(0.1, 10^{-4})$	(50)	0.46	(0.03)	0.50	(0.02)	4.28	(0.11)	2.39	(0.08)	1.90	(0.07)
VB$(10^{-4}, 0.1)$	(50)	0.42	(0.02)	0.60	(0.01)	4.63	(0.07)	1.86	(0.03)	2.77	(0.05)
VB$(10^{-4}, 10^{-4})$	(50)	0.42	(0.02)	0.60	(0.01)	4.62	(0.07)	1.85	(0.03)	2.76	(0.06)
GS$(0.1, 0.1)$	(50)	0.37	(0.02)	0.51	(0.01)	5.45	(0.07)	2.35	(0.09)	3.20	(0.03)
GS$(0.1, 10^{-4})$	(50)	0.38	(0.01)	0.51	(0.01)	5.47	(0.04)	2.26	(0.03)	3.22	(0.01)
GS$(10^{-4}, 0.1)$	(50)	0.36	(0.02)	0.49	(0.01)	5.73	(0.05)	2.41	(0.04)	3.31	(0.03)
GS$(10^{-4}, 10^{-4})$	(50)	0.37	(0.02)	0.49	(0.01)	5.74	(0.03)	2.42	(0.02)	3.32	(0.02)
EM	(40)	0.42	(0.03)	0.60	(0.02)	4.37	(0.14)	1.84	(0.07)	2.55	(0.08)
EM	(25)	0.46	(0.03)	0.56	(0.02)	**4.23**	(0.17)	2.05	(0.09)	2.19	(0.08)
EM	(10)	0.41	(0.01)	0.43	(0.01)	4.32	(0.04)	2.74	(0.03)	**1.58**	(0.05)

Table 1: Evaluation of models produced by the various estimators. The values of the Dirichlet prior parameters for α_x and α_y appear in the estimator name for the VB and GS estimators, and the number of hidden states is given in parentheses. Reported values are means over all runs, followed by standard deviations. 10 runs were performed for each of the EM and VB estimators, while 5 runs were performed for the GS estimators. Each EM and VB run consisted of 1,000 iterations, while each GS run consisted of 50,000 iterations. For the estimators with 10 runs, a 3-standard error 95% confidence interval is approximately the same as the standard deviation.

states are being mapped onto a single POS tag. This is also consistent with the fact that the cross-entropy $H(T|Y)$ of tags given hidden states is relatively low (i.e., given a hidden state, the tag is relatively predictable), while the cross-entropy $H(Y|T)$ is relatively high.

4 Bayesian estimation via Gibbs Sampling and Variational Bayes

A Bayesian estimator combines a likelihood term $\mathrm{P}(\mathbf{x}|\theta, \phi)$ and a prior $\mathrm{P}(\theta, \phi)$ to estimate the posterior probability of a model or hidden state sequence. We can use a Bayesian prior to bias our estimator towards models that generate more skewed distributions. Because HMMs (and PCFGs) are products of multinomials, Dirichlet distributions are a particularly natural choice for the priors since they are conjugate to multinomials, which simplifies both the mathematical and computational aspects of the problem. The precise form of the model we investigated is:

$$\begin{array}{rcll} \theta_y & \mid & \alpha_y & \sim & \mathrm{Dir}(\alpha_y) \\ \phi_y & \mid & \alpha_x & \sim & \mathrm{Dir}(\alpha_x) \\ y_i & \mid & y_{i-1} = y & \sim & \mathrm{Multi}(\theta_y) \\ x_i & \mid & y_i = y & \sim & \mathrm{Multi}(\phi_y) \end{array}$$

Informally, α_y controls the sparsity of the state-to-state transition probabilities while α_x controls the sparsity of the state-to-observation emission probabilities. As α_x approaches zero the prior strongly prefers models in which each hidden state emits as few words as possible. This captures the intuition that most word types only belong to one POS, since the minimum number of non-zero state-to-observation transitions occurs when each observation type is emitted from only one state. Similarly, as α_y approaches zero the state-to-state transitions become sparser.

There are two main techniques for Bayesian estimation of such models: Markov Chain Monte Carlo (MCMC) and Variational Bayes (VB). MCMC encompasses a broad range of sampling techniques, including component-wise Gibbs sampling, which is the MCMC technique we used here (Robert and Casella, 2004; Bishop, 2006). In general, MCMC techniques do not produce a single model that characterizes the posterior, but instead produce a stream of samples from the posterior. The application of MCMC techniques, including Gibbs sampling, to HMM inference problems is relatively well-known: see Besag (2004) for a tutorial introduction and Goldwater and Griffiths (2007) for an application of Gibbs sampling to HMM inference for semi-supervised and unsupervised POS tagging.

The Gibbs sampler produces state sequences $\mathbf{y}$ sampled from the posterior distribution:

$$\mathrm{P}(\mathbf{y}|\mathbf{x}, \alpha) \propto \int \mathrm{P}(\mathbf{x}, \mathbf{y}|\theta, \phi)\mathrm{P}(\theta|\alpha_y)\mathrm{P}(\phi|\alpha_x)\, d\theta\, d\phi$$

Because Dirichlet priors are conjugate to multinomials, it is possible to integrate out the model parameters θ and ϕ to yield the conditional distribution for y_i shown in Figure 4. For each observation x_i in turn, we resample its state y_i conditioned on the states $\mathbf{y}_{-i}$ of the other observations; eventually the distribution of state sequences converges to the desired posterior.

Each iteration of the Gibbs sampler is much faster than the Forward-Backward algorithm (both take time linear in the length of the string, but for an HMM with s hidden states, each iteration of the Gibbs sampler takes $O(s)$ time while each iteration of the Forward-Backward algorithm takes $O(s^2)$ time), so we ran 50,000 iterations of all samplers (which takes roughly the same elapsed time as 1,000 Forward-Backward iterations).

As can be seen from Table 1, the posterior state sequences we obtained are not particularly good. Further, when we examined how the posterior likelihoods varied with increasing iterations of Gibbs sampling, it became apparent that the likelihood was still increasing after 50,000 iterations. Moreover, when comparing posterior likelihoods from different runs with the same prior parameters but different random number seeds, none of the likelihoods crossed, which one would expect if the samplers had converged and were mixing well (Robert and Casella, 2004). Just as with EM, we experimented with a variety of annealing regimes, but were unable to find any which significantly improved accuracy or posterior likelihood.

We also experimented with evaluating state sequences found using maximum posterior decoding (i.e., model parameters are estimated from the posterior sample, and used to perform maximum posterior decoding) rather than the samples from the posterior produced by the Gibbs sampler. We found that the maximum posterior decoding sequences usually scored higher than the posterior samples, but the scores converged after the first thousand iterations. Since the posterior samples are produced as a by-product of Gibbs sampling while maximum posterior decoding requires an additional time consuming step that does not have much impact on scores, we used the posterior samples to produce the results in Table 1.

In contrast to MCMC, Variational Bayesian inference attempts to find the function $Q(\mathbf{y}, \theta, \phi)$ that minimizes an upper bound of the negative log likelihood (Jordan et al., 1999):

$$\begin{aligned} & -\log \mathrm{P}(\mathbf{x}) \\ = \; & -\log \int Q(\mathbf{y}, \theta, \phi) \frac{\mathrm{P}(\mathbf{x}, \mathbf{y}, \theta, \phi)}{Q(\mathbf{y}, \theta, \phi)}\, d\mathbf{y}\, d\theta\, d\phi \\ \leq & -\int Q(\mathbf{y}, \theta, \phi) \log \frac{\mathrm{P}(\mathbf{x}, \mathbf{y}, \theta, \phi)}{Q(\mathbf{y}, \theta, \phi)}\, d\mathbf{y}\, d\theta\, d\phi (3) \end{aligned}$$

The upper bound in (3) is called the *Variational Free Energy*. We make a "mean-field" assumption that the posterior can be well approximated by a factorized model Q in which the state sequence $\mathbf{y}$ does not covary with the model parameters θ, ϕ (this will be true if, for example, there is sufficient data that the posterior distribution has a peaked mode):

$$\mathrm{P}(\mathbf{x}, \mathbf{y}, \theta, \phi) \;\approx\; Q(\mathbf{y}, \theta, \phi) \;=\; Q_1(\mathbf{y})Q_2(\theta, \phi)$$

The calculus of variations is used to minimize the KL divergence between the desired posterior distribution and the factorized approximation. It turns out that if the likelihood and conjugate prior belong to exponential families then the optimal Q_1 and Q_2 do too, and there is an EM-like iterative procedure that finds locally-optimal model parameters (Bishop, 2006).

This procedure is especially attractive for HMM inference, since it involves only a minor modification to the M-step of the Forward-Backward algorithm. MacKay (1997) and Beal (2003) describe Variational Bayesian (VB) inference for HMMs in detail, and Kurihara and Sato (2006) describe VB for PCFGs (which only involves a minor modification to the M-step of the Inside-Outside algorithm). Specifically, the E-step for VB inference for HMMs is the same as in EM, while the M-step is as follows:

$$\begin{aligned} \tilde{\theta}^{(\ell+1)}_{y'|y} &= f(\mathrm{E}[n_{y',y}] + \alpha_y)/f(\mathrm{E}[n_y] + s\alpha_y) \quad (4) \\ \tilde{\phi}^{(\ell+1)}_{x|y} &= f(\mathrm{E}[n_{x,y}] + \alpha_x)/f(\mathrm{E}[n_y] + m\alpha_x) \\ f(v) &= \exp(\psi(v)) \\ \psi(v) &= (v > 7) \; ? \; g(v - \tfrac{1}{2}) : (\psi(v+1) - 1)/v \\ g(x) &\approx \log(x) + 0.04167x^{-2} + 0.00729x^{-4} \\ & \quad +0.00384x^{-6} - 0.00413x^{-8} \ldots \quad (5) \end{aligned}$$

$$\mathrm{P}(y_i|\mathbf{x}, \mathbf{y}_{-i}, \alpha) \quad \propto \quad \left(\frac{n_{x_i,y_i} + \alpha_x}{n_{y_i} + m\alpha_x}\right) \left(\frac{n_{y_i,y_{i-1}} + \alpha_y}{n_{y_{i-1}} + s\alpha_y}\right) \left(\frac{n_{y_{i+1},y_i} + \mathrm{I}(y_{i-1} = y_i = y_{i+1}) + \alpha_y}{n_{y_i} + \mathrm{I}(y_{i-1} = y_i)}\right)$$

Figure 4: The conditional distribution for state y_i used in the Gibbs sampler, which conditions on the states $\mathbf{y}_{-i}$ for all observations *except* x_i. Here m is the number of possible observations (i.e., the size of the vocabulary), s is the number of hidden states and $\mathrm{I}(\cdot)$ is the indicator function (i.e., equal to one if its argument is true and zero otherwise), $n_{x,y}$ is the number of times observation x occurs with state y, $n_{y',y}$ is the number of times state y' follows y, and n_y is the number of times state y occurs; these counts are from $(\mathbf{x}_{-i}, \mathbf{y}_{-i})$, i.e., excluding x_i and y_i.

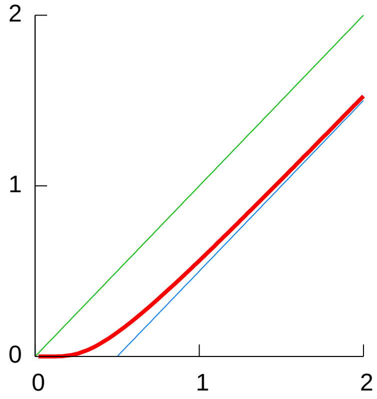

Figure 5: The scaling function $y = f(x) = \exp\psi(x)$ (curved line), which is bounded above by the line $y = x$ and below by the line $y = x - 0.5$.

where ψ is the *digamma function* (the derivative of the log gamma function; (5) gives an asymptotic approximation), and the remaining quantities are just as in the EM updates (2), i.e., $n_{x,y}$ is the number of times observation x occurs with state y, $n_{y',y}$ is the number of times state y' follows y, n_y is the number of occurences of state y, s is the number of hidden states and m is the number of observations; all expectations are taken with respect to the variational parameters $(\tilde{\theta}^{(\ell)}, \tilde{\phi}^{(\ell)})$.

A comparison between (4) and (2) reveals two differences between the EM and VB updates. First, the Dirichlet prior parameters α are added to the expected counts. Second, these posterior counts (which are in fact parameters of the Dirichlet posterior Q_2) are passed through the function $f(v) = \exp\psi(v)$, which is plotted in Figure 5. When $v \gg 0$, $f(v) \approx v - 0.5$, so roughly speaking, VB for multinomials involves adding $\alpha - 0.5$ to the expected counts when they are much larger than zero, where α is the Dirichlet prior parameter. Thus VB can be viewed as a more principled version of the well-known ad hoc technique for approximating Bayesian estimation with EM that involves adding $\alpha - 1$ to the expected counts. However, in the ad hoc approach the expected count plus $\alpha - 1$ may be less than zero, resulting in a value of zero for the corresponding parameter (Johnson et al., 2007; Goldwater and Griffiths, 2007). VB avoids this problem because $f(v)$ is always positive when $v > 0$, even when v is small. Note that because the counts are passed through f, the updated values for $\tilde{\theta}$ and $\tilde{\phi}$ in (4) are in general *not* normalized; this is because the variational free energy is only an upper bound on the negative log likelihood (Beal, 2003).

We found that in general VB performed much better than GS. Computationally it is very similar to EM, and each iteration takes essentially the same time as an EM iteration. Again, we experimented with annealing in the hope of speeding convergence, but could not find an annealing schedule that significantly lowered the variational free energy (the quantity that VB optimizes). While we had hoped that the Bayesian prior would bias VB toward a common solution, we found the same sensitivity to initial conditions as we found with EM, so just as for EM, we ran the estimator for 1,000 iterations with 10 different random initializations for each combination of prior parameters. Table 1 presents the results of VB runs with several different values for the Dirichlet prior parameters. Interestingly, we obtained our best performance on 1-to-1 accuracy when the Dirchlet prior $\alpha_x = 0.1$, a relatively large number, but best performance on many-to-1 accuracy was achieved with a much lower value for the Dirichlet prior, namely $\alpha_x = 10^{-4}$. The Dirichlet prior α_y that controls sparsity of the state-to-state transitions had little effect on the results. We did not have computational resources to fully explore other values for the prior (a set of 10 runs for one set of parameter values takes 25 computer days).

As Figure 3 shows, VB can produce distributions of hidden states that are peaked in the same way that POS tags are. In fact, with the priors used here, VB produces state sequences in which only a subset of the possible HMM states are in fact assigned to observations. This shows that rather than fixing the number of hidden states in advance, the Bayesian prior can determine the number of states; this idea is more fully developed in the *infinite HMM* of Beal et al. (2002) and Teh et al. (2006).

5 Reducing the number of hidden states

EM already performs well in terms of the many-to-1 accuracy, but we wondered if there might be some way to improve its 1-to-1 accuracy and VI score. In section 3 we suggested that one reason for its poor performance in these evaluations is that the distributions of hidden states it finds tend to be fairly flat, compared to the empirical distribution of POS tags. As section 4 showed, a suitable Bayesian prior can bias the estimator towards more peaked distributions, but we wondered if there might be a simpler way of achieving the same result.

We experimented with dramatic reductions in the number of hidden states in the HMMs estimated by EM. This should force the hidden states to be more densely populated and improve 1-to-1 accuracy, even though this means that there will be no hidden states that can possibly map onto the less frequent POS tags (i.e., we will get these words wrong). In effect, we abandon the low-frequency POS tags in the hope of improving the 1-to-1 accuracy of the high-frequency tags.

As Table 1 shows, this markedly improves both the 1-to-1 accuracy and the VI score. A 25-state HMM estimated by EM performs effectively as well as the best VB model in terms of both 1-to-1 accuracy and VI score, and runs 4 times faster because it has only half the number of hidden states.

6 Conclusion and future work

This paper studied why EM seems to do so badly in HMM estimation for unsupervised POS tagging. In fact, we found that it doesn't do so badly at all: the bitag HMM estimated by EM achieves a mean 1-to-1 tagging accuracy of 40%, which is approximately the same as the 41.3% reported by (Haghighi and Klein, 2006) for their sophisticated MRF model.

Then we noted the distribution of words to hidden states found by EM is relatively uniform, compared to the distribution of words to POS tags in the evaluation corpus. This provides an explanation of why the many-to-1 accuracy of EM is so high while the 1-to-1 accuracy and VI of EM is comparatively low. We showed that either by using a suitable Bayesian prior or by simply reducing the number of hidden states it is possible to significantly improve both the 1-to-1 accuracy and the VI score, achieving a 1-to-1 tagging accuracy of 46%.

We also showed that EM and other estimators take much longer to converge than usually thought, and often require several hundred iterations to achieve optimal performance. We also found that there is considerable variance in the performance of all of these estimators, so in general multiple runs from different random starting points are necessary in order to evaluate an estimator's performance.

Finally, there may be more sophisticated ways of improving the 1-to-1 accuracy and VI score than the relatively crude methods used here that primarily reduce the number of available states. For example, we might obtain better performance by using EM to infer an HMM with a large number of states, and then using some kind of distributional clustering to group similar HMM states; these clusters, rather than the underlying states, would be interpreted as the POS tag labels. Also, the Bayesian framework permits a wide variety of different priors besides Dirichlet priors explored here. For example, it should be possible to encode linguistic knowledge such markedness preferences in a prior, and there are other linguistically uninformative priors, such the "entropic priors" of Brand (1999), that may be worth exploring.

Acknowledgements

I would like to thank Microsoft Research for providing an excellent environment in which to conduct this work, and my friends and colleagues at Microsoft Research, especially Bob Moore, Chris Quirk and Kristina Toutanova, for their helpful comments on this paper.

References

Michele Banko and Robert C. Moore. 2004. Part of speech tagging in context. In *Proceedings, 20th International Conference on Computational Linguistics (Coling 2004)*, pages 556–561, Geneva, Switzerland.

M.J. Beal, Z. Ghahramani, and C.E. Rasmussen. 2002. The infinite Hidden Markov Model. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 577–584. The MIT Press.

Matthew J. Beal. 2003. *Variational Algorithms for Approximate Bayesian Inference*. Ph.D. thesis, Gatsby Computational Neuroscience unit, University College London.

Julian Besag. 2004. An introduction to Markov Chain Monte Carlo methods. In Mark Johnson, Sanjeev P. Khudanpur, Mari Ostendorf, and Roni Rosenfeld, editors, *Mathematical Foundations of Speech and Language Processing*, pages 247–270. Springer, New York.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

M. Brand. 1999. An entropic estimator for structure discovery. *Advances in Neural Information Processing Systems*, 11:723–729.

Glenn Carroll and Eugene Charniak. 1992. Two experiments on learning probabilistic dependency grammars from corpora. In *Proceedings of the AAAI Workshop on Statistically-Based Natural Language Processing Techniques*, San Jose, CA.

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 59–66. Association for Computational Linguistics.

Victoria Fromkin, editor. 2001. *Linguistics: An Introduction to Linguistic Theory*. Blackwell, Oxford, UK.

Sharon Goldwater and Tom Griffiths. 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Joshua Goodman. 2001. A bit of progress in language modeling. *Computer Speech and Language*, 14:403–434.

Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 320–327, New York City, USA, June. Association for Computational Linguistics.

Frederick Jelinek. 1997. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts.

Mark Johnson, Tom Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146, Rochester, New York. Association for Computational Linguistics.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Sau. 1999. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Dan Klein. 2005. *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis, Stanford University.

Kenichi Kurihara and Taisuke Sato. 2006. Variational Bayesian grammar induction for natural language. In *8th International Colloquium on Grammatical Inference*.

David J.C. MacKay. 1997. Ensemble learning for hidden Markov models. Technical report, Cavendish Laboratory, Cambridge.

Chris Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Michell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Marina Meilă. 2003. Comparing clusterings by the variation of information. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *COLT 2003: The Sixteenth Annual Conference on Learning Theory*, volume 2777 of *Lecture Notes in Computer Science*, pages 173–187. Springer.

Bernard Merialdo. 1994. Tagging English text with a probabilistic model. *Computational Linguistics*, 20:155–171.

M. Mitzenmacher. 2003. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2):226–251.

Christian P. Robert and George Casella. 2004. *Monte Carlo Statistical Methods*. Springer.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the*

Association for Computational Linguistics (ACL'05), pages 354–362, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Noah A. Smith. 2006. *Novel Estimation Methods for Unsupervised Discovery of Latent Structure in Natural Language Text*. Ph.D. thesis, Johns Hopkins University.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259.

Qin Iris Wang and Dale Schuurmans. 2005. Improved estimation for unsupervised part-of-speech tagging. In *Proceedings of the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE'2005)*, pages 219–224, Wuhan, China.