Collapsed Variational Bayesian Inference for Hidden Markov Models

Pengyu Wang
Department of Computer Science
University of Oxford
Oxford, OX1 3QD, United Kingdom
Pengyu.Wang@cs.ox.ac.uk

Phil Blunsom
Department of Computer Science
University of Oxford
Oxford, OX1 3QD, United Kingdom
Phil.Blunsom@cs.ox.ac.uk

Abstract

Approximate inference for Bayesian models is dominated by two approaches, variational Bayesian inference and Markov Chain Monte Carlo. Both approaches have their own advantages and disadvantages, and they can complement each other. Recently researchers have proposed collapsed variational Bayesian inference to combine the advantages of both. Such inference methods have been successful in several models whose hidden variables are conditionally independent given the parameters. In this paper we propose two collapsed variational Bayesian inference algorithms for hidden Markov models, a popular framework for representing time series data. We validate our algorithms on the natural language processing task of unsupervised part-of-speech induction, showing that they are both more computationally efficient than sampling, and more accurate than standard variational Bayesian inference for HMMs.

1 Introduction

Hidden Markov Models (HMMs) are widely used for representing sequential data in various fields including speech recognition, natural language processing, information retrieval, computer vision, bioinformatics and finance. The core theory of HMMs, together with the celebrated forward-backward (or Baum-Welch) algorithm was developed by Baum and colleagues (Baum and Petrie, 1966; Baum et al., 1970). As a simple but effective statistical tool, the popularity of HMMs soared in the following decade, yielding a variety of elaborations and applications, reviewed by Juang and Rabiner (1991). Smyth et al. (1997) expressed HMMs as Bayesian networks, which promoted the development of a number of Bayesian approaches (MacKay, 1997; Beal, 2003; Goldwater and Griffiths, 2007).

Variational Bayesian inference (VB) (MacKay, 1997; Beal, 2003) for HMMs seeks to minimise the divergence between the true posterior and an approximation in which the parameters and hidden variables are assumed independent. This strong assumption allows for an efficient iterative solution, but it can often lead to poor approximations. Alternatively, collapsed Gibbs sampling (CGS) for HMMs by Goldwater and Griffiths (2007) integrates out the parameters, and draws samples for hidden variables in turn from the true posterior. In theory, CGS reaches the true posterior after convergence. In practice it is notoriously difficult to assess the convergence of samplers, and mixing is slow for distributions with tightly coupled latent variables like the HMM. It remains a challenge to develop algorithms that are both accurate and efficient, especially for large scale problems in our application domain of natural language processing.

Recently Teh et al. (2007) and Sung et al. (2008) suggest a third class of algorithms: collapsed variational Bayesian inference (CVB), which applies variational inference in the same collapsed space as CGS. Integrating out the parameters induces dependencies which spread over many hidden variables, and thus the dependency between any two hidden variables is very weak. Following the collapsing step, the hidden variables are assumed to be independent and mean field inference is applied.

Sung et al. (2008) studied CVB inference in the context of the general conjugate-exponential family. Nevertheless, one cannot derive CVB for a particular model based on this general result. Teh et al. (2007) successfully applied CVB inference to latent Dirichlet allocation (LDA), a popular framework for topic

Appearing in Proceedings of the 16^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. Volume 31 of JMLR: W&CP 31.

modeling, and also suggested the usage of CVB in a wider class of discrete graphical models, including HMMs. To date CVB has not been extended to models that have time series dependencies (e.g. HMMs) or structural dependencies (e.g. probabilistic context free grammars (PCFGs)). For such models, there exist strong dependencies among hidden variables even in the collapsed space. In this context the independence assumption, required by the mean field method, is not an optimal choice.

In this paper we propose two CVB algorithms for HMMs. The first algorithm assumes independence among the hidden variables in the collapsed space, as was done for LDA (Teh et al., 2007). The second algorithm keeps the strong dependencies in the original model and neglects the weak dependencies induced by marginalizing out the parameters. Because of such weak assumptions the second CVB algorithm can approximate the true posterior very closely. Our experiments show that both algorithms surpass VB inference, while maintaining the same computational complexity. The second algorithm outperforms CGS as the size of the data increases, with an order of magnitude less training time.

2 Approximate inference for HMMs

A HMM models a sequence of observations $\mathbf{x} = x_1, x_2, ..., x_T$, together with a sequence of hidden states $\mathbf{z} = z_0, z_1, z_2, ..., z_T$, which is generated by a first order Markov process. Each observation $x_t \in \{1, ..., W\}$ is emitted by the corresponding hidden state $z_t \in \{1, ..., K\}$ at time step t. For convenience, we let $z_0 = s$, the start state, which is also included in the hidden state space. The set of parameters θ for a HMM consists of a transition matrix A and en emission matrix B. Let $A_{k,k'}$ denote the probability of transitioning into state k' from state k, and $B_{k,w}$ denote the probability of emitting observation w from hidden state k. Symmetric Dirichlet priors with hyperparameters α and β are placed on each of $A_k = \{A_{k,k'}\}$ and $B_k = \{B_{k,w}\}$, respectively. The joint probability of the parameters and variables is:

$$p(\mathbf{x}, \mathbf{z}, \theta|\alpha, \beta) = p(A|\alpha)p(B|\beta)\prod_{t=0}^{T-1} p(z_{t+1}|z_t, A)p(x_{t+1}|z_{t+1}, B) \quad (1)$$

2.1 Variational Bayesian inference

The standard Baum-Welch training algorithm (Baum et al., 1970) for a HMM is a special case of a general class of algorithms, namely expectation-maximization (EM) (Dempster et al., 1977), which can be further considered as a subclass of VB. VB inference lower bounds the log marginal likelihood of the data $\log p(\mathbf{x})$ by using the negative variational free energy.

$$\begin{aligned}\log p(\mathbf{x}|\alpha, \beta) &\geq -\mathcal{F}(q(\mathbf{z}, \theta)) \\ &= \mathbb{E}_{q(\mathbf{z},\theta)}[\log p(\mathbf{x}, \mathbf{z}, \theta|\alpha, \beta)] - \mathbb{E}_{q(\mathbf{z},\theta)}[\log q(\mathbf{z}, \theta)]\end{aligned} \quad (2)$$

with $q(\mathbf{z}, \theta)$ an approximate posterior, and $q(\mathbf{z}, \theta)$ is factorized by assuming independence between parameters and latent variables:

$$q(\mathbf{z}, \theta) \approx q(\mathbf{z})q(\theta) \quad (3)$$

Maximizing $-\mathcal{F}(q(\mathbf{z}, \theta))$ updates $q(\mathbf{z})$ and $q(\theta)$ in turn. For the Baum-Welch algorithm using maximum likelihood estimation, $q(\theta)$ degenerates, i.e. $\delta(\theta = \theta^\star)$.

$$\text{E step: } q(\mathbf{z}) \propto \exp(\mathbb{E}_{q(\theta)}[\log p(\mathbf{x}, \mathbf{z}, \theta)]) \quad (4)$$

$$\text{M step: } \theta^\star = \underset{\theta}{\operatorname{argmax}}\, p(\mathbf{x}, \mathbf{z}, \theta) \quad (5)$$

Solving the E step yields

$$\begin{aligned}q(\mathbf{z}) \propto& p(\mathbf{x}, \mathbf{z}|\theta^\star) \\ =& \prod_{t=0}^{T-1} p(z_{t+1}|z_t, A^\star)p(x_{t+1}|z_{t+1}, B^\star) \\ =& \prod_{t=0}^{T-1} A^\star_{z_t,z_{t+1}} B^\star_{z_{t+1},x_{t+1}}\end{aligned} \quad (6)$$

The distribution over the whole sequence $q(\mathbf{z})$ is intractable, dynamic programming tricks are carried out to compute $p(z_t, \mathbf{x}|\theta^\star)$ and $p(z_t, z_{t+1}, \mathbf{x}|\theta^\star)$.

$$p(z_t, \mathbf{x}|\theta^\star) \propto \alpha_t(z_t)\beta_t(z_t) \quad (7)$$

$$p(z_t, z_{t+1}, \mathbf{x}|\theta^\star) \propto \alpha_t(z_t)A^\star_{z_t,z_{t+1}}B^\star_{z_{t+1},x_{t+1}}\beta_{t+1}(z_{t+1}) \quad (8)$$

where $\alpha_t(k)$ is the forward probability of being in state k at time step t, given observations before and include t; $\beta_t(k)$ is the backward probability of seeing observations after t, given state is k at time step t. Both are computed recursively.

Solving the M step yields

$$A^\star_{k,k'} = \frac{\sum_{t=0}^{T-1} q(z_t = k, z_{t+1} = k')}{\sum_{t=0}^{T-1} q(z_t = k)} \quad (9)$$

$$B^\star_{k,w} = \frac{\sum_{t=1}^{T} q(z_t = k)\delta(x_t = w)}{\sum_{t=1}^{T} q(z_t = k)} \quad (10)$$

VB inference generalizes EM by putting no restrictions on the parametric form of $q(\theta)$. Thus, the update in the M step becomes

$$q(\theta) \propto \exp(\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}, \mathbf{z}, \theta|\alpha, \beta)]) \quad (11)$$

$$p(z_t = k|\mathbf{x}, \mathbf{z}^{\neg t}, \alpha, \beta) \propto \frac{C^{\neg t}_{k,w} + \beta}{C^{\neg t}_{k,\cdot} + W\beta} \cdot \frac{C^{\neg t}_{z_{t-1},k} + \alpha}{C^{\neg t}_{z_{t-1},\cdot} + K\alpha} \cdot \frac{C^{\neg t}_{k,z_{t+1}} + \alpha + \delta(z_{t-1} = k = z_{t+1})}{C^{\neg t}_{k,\cdot} + K\alpha + \delta(z_{t-1} = k)}$$

Figure 1: The conditional distribution for a single hidden state z_i in the collapsed Gibbs sampler, conditioned on all other hidden states $\mathbf{z}^{\neg t}$. $C^{\neg i}$ is the count that does not include z_i, w is the observation at time step t, W is the size of observation space, and K is the size of hidden state space. δ is the standard indicator function.

Solving the above equation results in Dirichlet distributions with updated hyperparameters. Equivalently, Beal (2003) suggested the mean parameters $\tilde{\theta}$ instead. This involves only a minor change in the M step:

$$\tilde{A}_{k,k'} = \frac{f(\sum_{t=0}^{T-1} q(z_t = k, z_{t+1} = k') + \alpha)}{f(\sum_{t=0}^{T-1} q(z_t = k) + K\alpha)}$$
$$\tilde{B}_{k,w} = \frac{f(\sum_{t=1}^{T} q(z_t = k)\delta(x_t = w) + \beta)}{f(\sum_{t=1}^{T} q(z_t = k) + W\beta)}$$
$$f(x) = \exp(\Psi(x))$$

where $\Psi(x) = \frac{\partial \Gamma(x)}{\partial x}$ is the digamma function.

Ignoring how fluctuations in θ induce fluctuations in $\mathbf{z}$ (and vice-versa) allows for analytic iterations, and both EM and VB inference algorithms are efficient and easy to implement. Nevertheless, the independence assumption may potentially lead to very inaccurate estimations. The parameters and latent variables are strongly dependent in the true posterior $p(\mathbf{z}, \theta|\mathbf{x}, \alpha, \beta)$, which is proportional to the joint distribution in (1). As we shall see in the following, CGS and CVB model the dependencies between parameters and hidden variables in an exact fashion.

2.2 Collapsed Gibbs sampling

The collapsed Gibbs sampler produces a hidden state sequence $\mathbf{z}$ sampled from the posterior distribution

$$p(\mathbf{z}|\mathbf{x}, \alpha, \beta) = \int p(\mathbf{z}, \mathbf{x}|\theta)p(\theta|\alpha, \beta)d\theta \qquad (12)$$

Because Dirichlet priors are conjugate to discrete distributions, it is possible to integrate out the model parameters θ to yield the conditional distribution for z_i shown in Figure 1. The derivation is quite standard by following the tutorial (Resnik and Hardisty, 2010). It also appeared in Goldwater and Griffiths (2007), and Gao and Johnson (2008).

CGS does not make any independence assumptions between parameters and hidden variables, and draws samples from the true posterior. However, as with other MCMC samplers, it is often hard to assess convergence, and one needs to set the number of samples and the burn-in period based on experience. In practice, one often draws as many samples as possible (within the limited time frame) to reduce sampling variance, and thus it is much less efficient than EM and VB.

Griffiths and Steyvers (2004) observed that the CGS for LDA converged relatively quickly. In LDA, the conditional distribution for the currently updating variable depends on other variables only through the counts, i.e. the dependency on any particular other variable is very small. Hence quick convergence is to be expected. For HMMs the conditional distribution for z_t in Figure 1 depends on the states of the previous hidden variable (z_{t-1}) and the next hidden variable (z_{t+1}), as well as the global counts. Such strong dependencies makes CGS for HMMs much slower to converge (Gao and Johnson, 2008).

3 Collapsed variational inference for i.i.d. hidden variables

The rapid convergence of CGS for LDA indicates that VB in the collapsed space is likely to be effective. For any independent and identically distributed models,[1] collapsing the parameters induces only weak dependencies among the hidden variables. The sum of the dependencies is decisive, but any particular dependency is tiny, especially for large data sets. This fits exactly with the assumptions underlying mean field theory. The currently updating variable relies on the field (i.e. summary statistics), through which it interacts with other variables. As the influence from any single variable on the field is small we may expect mean field updates in the collapsed space to be accurate.

Formally, CVB models the dependencies between parameters and hidden variables in an exact fashion.

$$q(\mathbf{z}, \theta) = q(\mathbf{z})q(\theta|\mathbf{z}) \qquad (13)$$

The mean field method requires independent variables, and thus the induced weak dependencies among hid-

[1] We define a model to be i.i.d., if any two hidden variables are conditionally independent given the parameters. LDA and mixture models are typical examples. HMMs are not i.i.d., as each hidden variable is dependent on the previous one given the parameters.

den variables are neglected.

$$q(\mathbf{z}) \approx \prod_{t=1}^{T} q(z_t) \tag{14}$$

The negative variational free energy becomes:

$$\begin{aligned}-\mathcal{F}(q(\mathbf{z})q(\theta|\mathbf{z})) =& \mathbb{E}_{q(\mathbf{z})q(\theta|\mathbf{z})}[\log p(\mathbf{x},\mathbf{z},\theta) - \log q(\mathbf{z},\theta)] \\ =& \mathbb{E}_{q(\mathbf{z})}[\mathbb{E}_{q(\theta|\mathbf{z})}[\log \frac{p(\mathbf{x},\mathbf{z},\theta)}{q(\theta|\mathbf{z})}] - \log q(\mathbf{z})]\end{aligned} \tag{15}$$

Maximizing $-\mathcal{F}(q(\mathbf{z})q(\theta|\mathbf{z}))$ updates $q(\theta|\mathbf{z})$ and $q(\mathbf{z})$ in turn. We set $q(\theta|\mathbf{z})$ equal to the true posterior:

$$\begin{aligned}-\mathcal{F}(q(\mathbf{z})p(\theta|\mathbf{x},\mathbf{z})) =& \mathbb{E}_{q(\mathbf{z})}[\mathbb{E}_{p(\theta|\mathbf{x},\mathbf{z})}[\log \frac{p(\mathbf{x},\mathbf{z},\theta)}{p(\theta|\mathbf{x},\mathbf{z})}] \\ & - \log q(\mathbf{z})] \\ =& \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z})]\end{aligned} \tag{16}$$

The mean field update for each $q(z_t)$ in $q(\mathbf{z})$ is:

$$\begin{aligned}q(z_t) \propto& \exp(\mathbb{E}_{q(\mathbf{z}^{\neg t})}[\log p(\mathbf{x},\mathbf{z})]) \\ \propto& \exp(\mathbb{E}_{q(\mathbf{z}^{\neg t})}[\log p(z_t|\mathbf{x},\mathbf{z}^{\neg t})])\end{aligned} \tag{17}$$

The term $p(z_t|\mathbf{x},\mathbf{z}^{\neg t})$ is exactly the conditional distribution for z_t, up to a normalization constant, used in CGS. It is easy to verify that the conditional distribution from Figure 1 can be substituted into (17). This suggests a systematic way to obtain CVB updates for any i.i.d. model for which the CGS posterior has already been derived.

4 Collapsed variational inference for HMMs

For HMMs with time series dependencies each hidden variable is dependent on the previous one given the parameters. In the collapsed space, a hidden variable strongly depends on both the previous and next variables, and weakly depends on others through their counts. We describe two CVB inference algorithms for HMMs which make different assumptions about the variational distribution $q(\mathbf{z})$.

4.1 Algorithm 1

The first CVB algorithm for HMMs follows the theory for i.i.d. models developed in the previous section by assuming that the hidden variables are mutually independent in the collapsed space. Although this is a strong assumption for HMMs, the independence between parameters and hidden variables in standard VB is also a strong assumption. It is not immediately apparent which assumption is weaker, and thus this CVB algorithm has the potential to lead to better approximations than the standard VB.

From (17), by substituting the conditional distribution for z_t in Figure 1, we get the first CVB algorithm for HMMs (Figure 2). The exact computation is too expensive for practical applications. We follow Teh et al. (2007) by using a Gaussian approximation, but use only the first order expansion of the Taylor approximation to compute each expected log count in Figure 2.[2] For example,

$$\mathbb{E}_{q(\mathbf{z}^{\neg t})}[\log(C^{\neg t}_{k,w} + \beta)] \approx \log(\mathbb{E}_{q(\mathbf{z}^{\neg t})}[C^{\neg t}_{k,w}] + \beta) \tag{18}$$

where $C^{\neg t}_{k,w} = \sum_{t' \neq t} \delta(x_{t'} = w)\delta(z_{t'} = k)$ is the sum of the independent Bernoulli variables $\delta(x_{t'} = w, z_{t'} = k)$ for all $t' \neq t$. That is:

$$\mathbb{E}_{q(\mathbf{z}^{\neg t})}[C^{\neg t}_{k,w}] = \sum_{t' \neq t} \delta(x_{t'} = w) q(z_{t'} = k) \tag{19}$$

Plugging (18) into Figure 2, we have the approximate solution as shown in Figure 3. The extreme similarity between CGS in Figure 1 (actual counts) and CVB in Figure 3 (expected counts) confirms that the first CVB algorithm is the mean field version of collapsed Gibbs sampling.

However, the strong independence assumption brings unforeseen challenges. The counts and delta functions involving z_{t-1} and z_{t+1} give rise to difficulties. Unlike in CGS, z_{t-1} and z_{t+1} are not fixed values, but distributions. The same scenario does not appear for i.i.d. models, as their CGS formula is a function of only the current updating variable z_t as well as the counts.

By utilizing the independence assumption in (14), the expected counts involving z_{t-1} and z_{t+1} in Figure 3 can be computed as follows,

$$\begin{aligned}&\mathbb{E}_{q(\mathbf{z}^{\neg t})}[C^{\neg t}_{z_{t-1},k}] \\ &\quad= \mathbb{E}_{q(z_{t-1})}[\mathbb{E}_{q(\mathbf{z}^{\neg(t-1,t)})}[C^{\neg(t-1,t)}_{z_{t-1},k} + C^{t-2,t-1}_{z_{t-1},k}]] \\ &\quad= \sum_{z_{t-1}=k'} q(z_{t-1} = k') \\ &\qquad\times (\mathbb{E}_{q(\mathbf{z}^{\neg(t-1,t)})}[C^{\neg(t-1,t)}_{k',k} + C^{t-2,t-1}_{k',k}]) \\ &\quad= \sum_{z_{t-1}=k'} q(z_{t-1} = k')\mathbb{E}_{q(\mathbf{z}^{\neg(t-1,t)})}[C^{\neg(t-1,t)}_{k',k}] \\ &\qquad+ q(z_{t-2} = k')q(z_{t-1} = k)\end{aligned} \tag{20}$$

The delta functions involving z_{t-1} and z_{t+1} in Figure 3

[2] Originally Teh et al. (2007) adopted the second order Taylor approximation which was shown to under perform the same algorithm with only the first order information in later work (Asuncion et al., 2009). Sato and Nakagawa (2012) argued that this was caused by a zero-forcing effect.

$$q(z_t = k) \propto \frac{\exp(\mathbb{E}_{q(\mathbf{z}^{\neg t})}[\log(C^{\neg t}_{k,w} + \beta) + \log(C^{\neg t}_{z_{t-1},k} + \alpha) + \log(C^{\neg t}_{k,z_{t+1}} + \alpha + \delta(z_{t-1} = k = z_{t+1}))])}{\exp(\mathbb{E}_{q(\mathbf{z}^{\neg t})}[\log(C^{\neg t}_{k,\cdot} + W\beta) + \log(C^{\neg t}_{z_{t-1},\cdot} + K\alpha) + \log(C^{\neg t}_{k,\cdot} + K\alpha + \delta(z_{t-1} = k))])}$$

Figure 2: The exact mean field update for the first CVB inference algorithm.

$$q(z_t = k) \propto \frac{\mathbb{E}_{q(\mathbf{z}^{\neg t})}[C^{\neg t}_{k,w}] + \beta}{\mathbb{E}_{q(\mathbf{z}^{\neg t})}[C^{\neg t}_{k,\cdot}] + W\beta} \cdot \frac{\mathbb{E}_{q(\mathbf{z}^{\neg t})}[C^{\neg t}_{z_{t-1},k}] + \alpha}{\mathbb{E}_{q(\mathbf{z}^{\neg t})}[C^{\neg t}_{z_{t-1},\cdot}] + K\alpha} \cdot \frac{\mathbb{E}_{q(\mathbf{z}^{\neg t})}[C^{\neg t}_{k,z_{t+1}}] + \alpha + \mathbb{E}_{q(\mathbf{z}^{\neg t})}[\delta(z_{t-1} = k = z_{t+1})]}{\mathbb{E}_{q(\mathbf{z}^{\neg t})}[C^{\neg t}_{k,\cdot}] + K\alpha + \mathbb{E}_{q(\mathbf{z}^{\neg t})}[\delta(z_{t-1} = k)]}$$

Figure 3: The update for the first CVB algorithm using a first order Taylor series approximation.

can be computed as follows,

$$\mathbb{E}_{q(\mathbf{z}^{\neg t})}[\delta(z_{t-1} = k = z_{t+1})] = q(z_{t-1} = k)q(z_{t+1} = k) \tag{21}$$

The implementation for the first CVB algorithm simply keeps track of the global expected counts $C_{k,w}$ and $C_{k',k}$, subtracting the expected counts for z_t (and z_{t-1} or z_{t+1} when needed). After updating $q(z_t)$, the mean counts around z_t are added back into the global counts. Each update of $q(z_t)$ has the computational complexity $O(K^2)$, which is same as EM and VB.

4.2 Algorithm 2

The strong independence assumption in Algorithm 1 has the potential to lead to inaccurate results. However in order to apply the mean field method one has to partition the latent variables into disjoint and independent groups.

Our investigation of CVB algorithms for HMMs is inspired by large scale applications in natural language processing. A common feature of those problems is that there are usually many short sequences (i.e. sentences), where each sequence is drawn i.i.d. from the same set of parameters. Therefore the collection of HMM sequences can be considered as an i.i.d. model with a shared set of parameters.

Let $\mathbf{x}_i$ be the i^{th} sequence of observations, and $\mathbf{z}_i$ be the i^{th} sequence of hidden states. Denote the number of sequences to be I. By using the derivation for i.i.d. models it is reasonable to assume that each hidden state sequence is independent of the others, since they are only weakly dependent in the collapsed space.

$$\begin{aligned} q(\mathbf{z}, \theta) =& q(\theta|\mathbf{z})q(\mathbf{z}) \\ \approx& q(\theta|\mathbf{z}) \prod_{i=1}^{I} q(\mathbf{z}_i) \end{aligned} \tag{22}$$

As with any i.i.d. model,

$$\begin{aligned} q(\mathbf{z}_i) &\propto \exp(\mathbb{E}_{q(\mathbf{z}^{\neg i})}[\log p(\mathbf{z}_i|\mathbf{x}, \mathbf{z}^{\neg i})]) \\ &\propto \exp(\mathbb{E}_{q(\mathbf{z}^{\neg i})}[\log p(\mathbf{x}_i, \mathbf{z}_i|\mathbf{x}^{\neg i}, \mathbf{z}^{\neg i})]) \end{aligned} \tag{23}$$

The challenge is to compute the term $p(\mathbf{x}_i, \mathbf{z}_i|\mathbf{x}^{\neg i}, \mathbf{z}^{\neg i})$. The exact computation includes expensive non-Markov delta functions, as shown in Figure 4. We approximate by assuming that hidden variables within a sequence only exhibit first order Markov dependencies and output independence.

$$\begin{aligned} &p(\mathbf{x}_i, \mathbf{z}_i|\mathbf{x}^{\neg i}, \mathbf{z}^{\neg i}) \\ &\quad\approx \prod_{t=0}^{T-1} p(z_{i,t+1}|z_{i,t}, \mathbf{x}^{\neg i}, \mathbf{z}^{\neg i}) p(x_{i,t+1}|z_{i,t+1}, \mathbf{x}^{\neg i}, \mathbf{z}^{\neg i}) \\ &\quad= \prod_{t=0}^{T-1} \frac{C^{\neg i}_{z_{i,t},z_{i,t+1}} + \alpha}{C^{\neg i}_{z_{i,t},\cdot} + K\alpha} \cdot \frac{C^{\neg i}_{z_{i,t+1},x_{i,t+1}} + \beta}{C^{\neg i}_{z_{i,t+1},\cdot} + W\beta} \end{aligned} \tag{24}$$

This approximation ignores the contributions from other parts of the i^{th} sequence to the global counts. Compared with contributions from all other sequences, we assume the impact of these local counts is small.

Substituting (24) into (23), and with the first order Taylor approximation,

$$q(\mathbf{z}_i) \approx \prod_{t=0}^{T-1} A^{\text{CVB}}_{z_{i,t},z_{i,t+1}} B^{\text{CVB}}_{z_{i,t+1},x_{i,t+1}} \tag{25}$$

where we define,

$$\begin{aligned} A^{\text{CVB}}_{z_{i,t},z_{i,t+1}} =& \frac{\mathbb{E}_{q(\mathbf{z}^{\neg i})}[C^{\neg i}_{z_{i,t},z_{i,t+1}}] + \alpha}{\mathbb{E}_{q(\mathbf{z}^{\neg i})}[C^{\neg i}_{z_{i,t},\cdot}] + K\alpha} \\ B^{\text{CVB}}_{z_{i,t+1},x_{i,t+1}} =& \frac{\mathbb{E}_{q(\mathbf{z}^{\neg i})}[C^{\neg i}_{z_{i,t+1},x_{i,t+1}}] + \beta}{\mathbb{E}_{q(\mathbf{z}^{\neg i})}[C^{\neg i}_{z_{i,t+1},\cdot}] + W\beta} \end{aligned}$$

The striking similarity between (6) and (25) suggests that the dynamic programming approach used in the EM and VB algorithms can be applied here. In the E step, the EM algorithm uses the maximum likelihood parameters $A^\star, B^\star$ from the M step; the VB algorithm uses the mean parameters $\tilde{A}, \tilde{B}$ from the M step; while the second CVB algorithm uses the parameters $A^{\text{CVB}}, B^{\text{CVB}}$ based on the expected counts from all other sequences. The main difference with EM and VB is that the parameters in CVB are dynamic,

$$p(\mathbf{x}_i, \mathbf{z}_i|\mathbf{x}^{\neg i}, \mathbf{z}^{\neg i}) = \prod_{t=0}^{T-1} \frac{C^{\neg i}_{z_{i,t},z_{i,t+1}} + \sum_{t'=0}^{t-1} \delta(z_{i,t'} = z_{i,t})\delta(z_{i,t'+1} = z_{i,t+1}) + \alpha}{C^{\neg i}_{z_{i,t},\cdot} + \sum_{t'=0}^{t-1} \delta(z_{i,t'} = z_{i,t}) + K\alpha} \cdot \frac{C^{\neg i}_{z_{i,t+1},x_{i,t+1}} + \sum_{t'=0}^{t-1} \delta(z_{i,t'+1} = z_{i,t+1})\delta(x_{i,t'+1} = x_{i,t+1}) + \beta}{C^{\neg i}_{z_{i,t+1},\cdot} + \sum_{t'=0}^{t-1} \delta(z_{i,t'+1} = z_{i,t+1}) + W\beta}$$

Figure 4: The exact computation of $p(\mathbf{x}_i, \mathbf{z}_i|\mathbf{x}^{\neg i}, \mathbf{z}^{\neg i})$ in the second CVB algorithm.

meaning that the parameters change after updating each sequence; whereas EM and VB batch update the parameters in the M step after processing all sequences in the E step.

In our described scenario (i.e. many short sequences), which is the norm in speech and language processing and also very common in other applications of HMMs, the assumptions made by our second CVB algorithm are weak. Therefore we might hope that its result will be very close to the true posterior.

5 Experiments

We validate the above inference algorithms for HMMs on the task of learning syntactic categories for words in text (part-of-speech tagging). We adopt the unsupervised formulation of Merialdo (1994): given a raw corpus and a tag dictionary that defines legal parts-of-speech for each word, tag each token in the corpus with the goal of maximizing accuracy against a reference tagged corpus. We experiment with simple bi-gram taggers with the aim of understanding the properties of our proposed inference algorithms, rather than building a state-of-the-art tagger (e.g. Berg-Kirkpatrick et al. (2010); Blunsom and Cohn (2011)).

Our data set is the Wall Street Journal (WSJ) treebank (Marcus et al., 1993). The tag dictionary is constructed by collecting all the tags found for each word type in the entire corpus. We conduct experiments for different corpus sizes, from 1K sentences to the entire treebank. For all the corpora, the percentages of ambiguous tokens is roughly 55% and the average number of tags per token in the dictionary is approximately 2.3. In later experiments we gradually relax the tag dictionary constraints until the tagging is fully ambiguous (i.e. fully unsupervised learning).

5.1 Varying the corpus size

In the first set of experiments the tagging accuracies of all the algorithms are compared for corpora of various sizes. Note that the EM algorithm optimizes likelihood ($p(\mathbf{x}|\theta)$), whereas the other algorithms optimize (a lower bound of) $p(\mathbf{x})$. The EM algorithm is included to serve as a benchmark. We run 50 iterations for the variational algorithms, and 20,000 iterations for CGS with the annealing scheme designed by Goldwater and Griffiths (2007) (temperature incrementally decreased from 2.0 to 0.08). Finally, each algorithm is run 10 times with different random initializations, and the hyperparameters α and $\beta \in [0.003, 0.01, 0.03, 0.1, 0.3, 1.0]$ are optimized on held-out data.

Table 1 presents the accuracies achieved by each of the algorithms on the various corpora. Both of the CVB algorithms outperform the standard VB algorithm, which in turn does not seem to have an advantage over EM. The first CVB algorithm surpasses VB by 2-3% , suggesting that in this experiment the seemingly strong hidden variable independence assumption proves to be weaker than the assumptions in standard VB. Furthermore, we expect the posterior to be highly peaked by using sparse priors, making the product of marginals a good approximation to the joint distribution. The second CVB algorithm yields the best results in most cases except for the 1K subset.

When CGS mixes quickly we would expect its performance to exceed all the other algorithms. From our experiments we see that this is the case for the smallest dataset, but as the data size increases CGS performs poorly. We find that the accuracies of the collapsed variational algorithms increase with the size of the data, countering the decreasing trend for CGS. In addition, the algorithms in the collapsed space have smaller variances than EM or standard VB.

Figure 5 shows the convergence rates for all the algorithms. The variational algorithms converge after approximately 15 iterations, whereas the accuracy of CGS is near its maximum with 10K iterations. All the variational algorithms have the same computational complexity ($O(TK^2)$), but their time usage varies in practice. Surprisingly, the most efficient method is not the EM algorithm as it has to be implemented in log space (or rescaled) to avoid underflow (Juang and Rabiner, 1991). VB requires computing expensive digamma functions, and the second CVB algorithm calculates dynamic parameters, hence both are slower than EM. Finally, CGS takes an order of magnitude more time than any of the deterministic algorithms.

		EM		VB		Algorithm 1		Algorithm 2		CGS	
Size	Random	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1K	65.1	81.0	1.2	79.2	1.3	82.9	0.2	84.2	0.2	**85.3**	0.5
2K	65.2	81.1	0.9	80.5	1.1	83.1	0.3	**85.5**	0.3	85.0	0.3
3K	65.1	81.1	0.8	80.5	1.0	83.1	0.2	**85.8**	0.3	85.0	0.2
5K	64.9	81.0	1.5	80.4	1.5	83.0	0.2	**85.6**	0.1	85.2	0.2
10K	64.7	81.4	1.7	80.7	1.2	83.4	0.2	**85.6**	0.1	85.0	0.2
All	64.8	81.4	0.9	81.4	1.1	83.7	0.1	**85.7**	0.1	84.6	0.1

Table 1: Tagging accuracies and standard deviations of 10 random runs on various corpus sizes with a complete tag dictionary. Viterbi tagging is used for EM and VB, whereas at each word position CGS chooses the tag with the maximum posterior. For Algorithm 1, both tagging methods achieve exactly the same results because of the independence assumption. For Algorithm 2, the maximum posterior tagging is slightly better (up to 0.1).

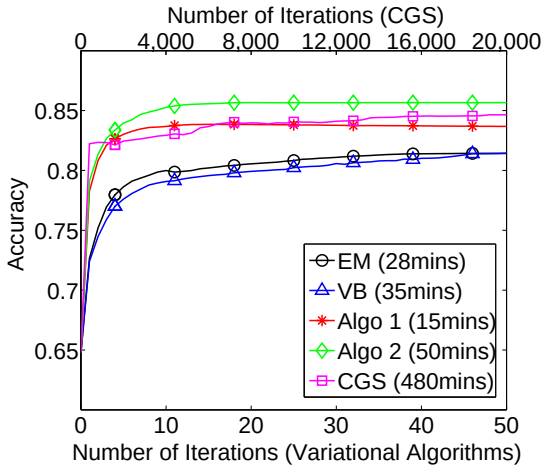

Figure 5: Accuracies averaged over 10 runs for the entire treebank with a complete tag dictionary. The variational algorithms were implemented in Python and run on an Intel Core i5 3.10GHZ computer with 4.0GB RAM. The CGS algorithm was implemented in C++.

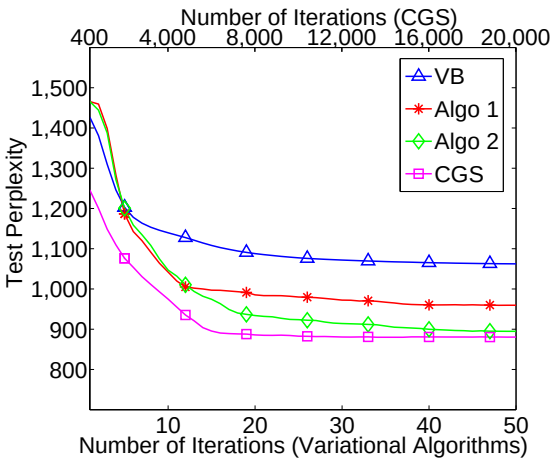

Figure 6: Perplexities averaged over 10 runs for the 1K data set without a tag dictionary.

5.2 Varying dictionary knowledge

In practice, it is not always possible to build a complete tag dictionary, especially for the infrequent words. We investigate the effects of reducing dictionary information. Following Smith and Eisner (2005), we randomly select 1K unlabelled sentences from the treebank for the training data[3]. We define a word type to be frequent if the word's tokens appear at least d times in the training corpus, otherwise it is infrequent. For frequent word types the standard tag dictionary is available; whereas for infrequent word types, all the tags are considered to be legal.

Table 2 presents the accuracies achieved by the algorithms at various ambiguity levels. Because of the small data set, the collapsed Gibbs sampler performs best in most cases, although somewhat surprisingly in some cases the second CVB algorithm outperforms CGS even in this small corpus. We find that with increasing ambiguity (approaching fully unsupervised learning), the margins between the standard VB and both of the CVB algorithms increase dramatically. In particular, when $d = 10$ (the average tags per token is 10.8, and the percentage of ambiguous tokens is 66%), the margin is as large as 13%.

[3]Small data sets significantly favor CGS. We hope that CGS can converge such that we can measure the margins between the results of the second CVB algorithm and CGS (i.e. close to the true posterior) in this and especially the next set of experiments.

5.3 Test perplexities

Without a tag dictionary the tag types are interchangeable and we have a label identifiability issue. Thus the tagging results cannot be evaluated directly against the reference tagged corpus. In this set of experiments, we randomly withhold 10% of the sentences from the data for testing, and use the remaining 90% for training. The algorithms are evaluated by their test perplexities (per token) on the withheld test set. We use $|\mathbf{x}_i|$ to denote the length of i^{th} sequence.

$$\text{perplexity}(\mathbf{x}_{\text{test}}) = 2^{\left(-\frac{\sum_i \log_2 p(\mathbf{x}_i)}{\sum_i |\mathbf{x}_i|}\right)} \tag{26}$$

			EM		VB		Algorithm 1		Algorithm 2		CGS	
d	tags/token	% ambig.	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1	2.3	54	81.0	1.2	79.2	1.3	82.9	0.2	84.2	0.2	**85.3**	0.5
2	7.8	63	69.6	1.5	69.6	1.6	73.8	0.4	77.9	0.4	**79.0**	0.4
3	10.8	66	64.7	1.7	61.1	1.6	65.1	0.4	**74.1**	0.3	72.5	0.7
5	14.9	71	54.7	2.9	54.3	2.0	64.7	0.4	**65.0**	0.5	64.2	0.9
10	20.8	77	43.5	2.4	43.6	2.1	51.2	0.7	53.3	0.5	**55.7**	0.6

Table 2: The accuracies and standard deviations are collected from 10 runs for the 1K data set with various incomplete dictionaries depending on values of d.

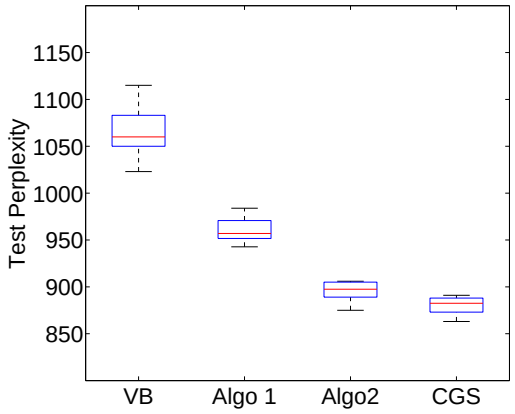

Figure 7: Perplexities at iteration 50 (20K for CGS) over 10 runs for the 1K data without a tag dictionary.

For the variational algorithms,

$$p(\mathbf{x}_i) = \sum_{\mathbf{z}_i} \prod_{t=0}^{T-1} \frac{\mathbb{E}_{q(\mathbf{z})}[C_{k',k}] + \alpha}{\mathbb{E}_{q(\mathbf{z})}[C_{k',\cdot}] + K\alpha} \cdot \frac{\mathbb{E}_{q(\mathbf{z})}[C_{k,w}] + \beta}{\mathbb{E}_{q(\mathbf{z})}[C_{k,\cdot}] + W\beta} \tag{27}$$

For CGS, given S samples from the posterior,[4]

$$p(\mathbf{x}_i) = \sum_{\mathbf{z}_i} \frac{1}{S} \sum_{s=1}^{S} \prod_{t=0}^{T-1} \frac{C^s_{k',k} + \alpha}{C^s_{k',\cdot} + K\alpha} \frac{C^s_{k,w} + \beta}{C^s_{k,\cdot} + W\beta} \tag{28}$$

where $z_{i,t} = k', z_{i,t+1} = k$ and $x_{i,t+1} = w$. Instead of using a grid search for the hyperparameters, we place Gamma priors on them ($\alpha \sim G(a,b), \beta \sim G(c,d)$), and use a fixed-point update (Minka, 2009), e.g.

$$\alpha' = \frac{a - 1 + \alpha \sum_{k'} \sum_{k} [\psi(C_{k',k} + \alpha) - \psi(\alpha)]}{b + K \sum_{k'} [\psi(C_{k',\cdot} + K\alpha) - \psi(K\alpha)]} \tag{29}$$

where $C_{k,k'}$ is the expected counts for the variational algorithms, and the actual counts for CGS. The resulting test perplexities in Figures 6 and 7 reconfirm the accuracy results achieved in the previous two sections. We find that both the CVB algorithms surpass the standard VB algorithm by large margins, and the second CVB algorithm is very close to CGS, which is assumed to have converged after 20,000 iterations.

[4]Annealing is not used in order to facilitate the calculation of perplexity.

6 Discussion

We have presented two collapsed variational Bayesian inference algorithms for hidden Markov models. Both algorithms are easy to implement, computationally efficient, and more accurate than the standard Baum-Welch and VB algorithms.

Our first CVB algorithm for the HMM makes strong assumptions about the independence of the hidden variables. The results indicate that these assumptions in the collapsed setting are superior to the parameter independence assumed in the standard VB algorithm. Notably, a common decoding algorithm for HMMs is to set each hidden variable to its maximum marginal probability assignment under the posterior. Thus the assumptions made by our first CVB algorithm naturally fit our decoding objective. Coupled with its efficiency, this suggest that this algorithm represents a practical trade-off between accuracy and scalability.

Our second CVB algorithm makes use of the common HMM scenario in which the number of sequences is large relative to their individual lengths. This allows us to accurately approximate the counts by discarding the small local contributions in favour of the contributions from all other sequences. Both theoretically and empirically the second algorithm is very accurate and appears to closely follow the true posterior. As the number of sequences increases this algorithm beats the collapsed Gibbs sampler with significantly less training time. Therefore the second algorithm achieves the same or better accuracy as CGS, with the efficiency of standard VB.

The results of our investigation indicate that the benefits of CVB may be more apparent for models which exhibit strong local coupling between hidden variables, than for the original LDA application. In this setting Gibbs sampling struggles to mix adequately and the time required to converge increases significantly. This suggests that many other Bayesian graphical models may also be amenable to CVB inference. In particular our work naturally extends to Bayesian models of probabilistic context free grammars (Johnson et al., 2007), and could be generalized to non-parametric HMMs (Beal et al., 2002).

References

Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press.

Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, 37(6): 1554–1563, 1966.

Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic function of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

Matthew Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, The Gatsby Computational Neuroscience Unit, University College London, 2003.

Matthew J. Beal, Zoubin Ghahramani, and Carl E. Rasmussen. The infinite hidden Markov model. In *Machine Learning*, pages 29–245. MIT Press, 2002.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590, Los Angeles, California, June 2010.

Phil Blunsom and Trevor Cohn. A hierarchical Pitman-Yor process HMM for unsupervised part of speech induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 865–874, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistics Society, Series B*, 39(1):1–38, 1977.

Jianfeng Gao and Mark Johnson. A comparison of Bayesian estimators for unsupervised hidden Markov model POS taggers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 344–352, Morristown, NJ, USA, 2008.

Sharon Goldwater and Tom Griffiths. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proc. of the 45th Annual Meeting of the ACL (ACL-2007)*, pages 744–751, Prague, Czech Republic, June 2007.

Thomas Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, April 2004. ISSN 0027-8424.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proc. of the 7th International Conference on Human Language Technology Research and 8th Annual Meeting of the NAACL (HLT-NAACL 2007)*, pages 139–146, Rochester, New York, April 2007.

Biing-Hwang Juang and Lawrence. R. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, 33(3):pp. 251–272, 1991.

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

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993. ISSN 0891-2017.

Bernard Merialdo. Tagging English text with a probabilistic model. *Comput. Linguist.*, 20(2):155–171, June 1994. ISSN 0891-2017.

Thomas P. Minka. Estimating a Dirichlet distribution. Technical report, 2009.

Philip Resnik and Eric Hardisty. Gibbs sampling for the uninitiated. Technical report, University of Maryland Computer Science Department, 2010.

Issei Sato and Hiroshi Nakagawa. Rethinking collapsed variational Bayes inference for LDA. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Noah A. Smith and Jason Eisner. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 354–362, Ann Arbor, Michigan, June 2005.

Padhraic Smyth, David Heckerman, and Michael I. Jordan. Probabilistic independence networks for hidden Markov probability models. *Neural Comput.*, 9(2):227–269, February 1997. ISSN 0899-7667.

Jaemo Sung, Zoubin Ghahramani, and Sung-Yang Bang. Latent-space variational Bayes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(12), December 2008.

Yee Whye Teh, David Newman, and Max Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *In Advances in Neural Information Processing Systems, volume 19*, 2007.