# Collapsed Variational Bayesian Inference for PCFGs

**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

This paper presents a collapsed variational Bayesian inference algorithm for PCFGs that has the advantages of two dominant Bayesian training algorithms for PCFGs, namely variational Bayesian inference and Markov chain Monte Carlo. In three kinds of experiments, we illustrate that our algorithm achieves close performance to the Hastings sampling algorithm while using an order of magnitude less training time; and outperforms the standard variational Bayesian inference and the EM algorithms with similar training time.

## 1 Introduction

Probabilistic context-free grammars (PCFGs) are commonly used in parsing and grammar induction systems (Johnson, 1998; Collins, 1999; Klein and Manning, 2003; Matsuzaki et al., 2005). The traditional method for estimating the parameters of PCFGs from terminal strings is the inside-outside (IO) algorithm (Baker, 1979). As a special instance of the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), based on the principle of maximum-likelihood estimation (MLE), the standard IO algorithm learns relatively uniform probability distributions for grammars, while the true distributions can be highly skewed (Johnson et al., 2007). In order to encourage sparse grammars and avoid overfitting, recent research for training PCFGs has drifted away from MLE in favor of Bayesian inference algorithms that make either deterministic or stochastic approximations (Kurihara and Sato, 2006; Johnson et al., 2006; Johnson et al., 2007).

Variational Bayesian inference (VB) (Kurihara and Sato, 2006) for PCFGs extends EM and places no constraints when updating parameters in the M step. By minimising the divergence between the true posterior and an approximate one in which the strong dependencies between the parameters and latent variables are broken, this deterministic algorithm efficiently converges to an inaccurate and only locally optimal solution like EM. Alternatively, Johnson et al. (2007) proposed two Markov Chain Monte Carlo algorithms for PCFGs that can reach the true posterior after convergence. However, it is often difficult to diagnose a sampler's convergence, and mixing is notoriously slow for distributions with tightly coupled hidden variables such as PCFGs, especially when the data sets are large. Therefore, there remains a challenge for more efficient, but also accurate and deterministic inference algorithms for PCFGs.

In this paper, we present a collapsed variational Bayesian inference (CVB) algorithm for PCFGs. It has the same computational complexity as the standard variational Bayesian inference, but offers almost the same performance as the stochastic algorithms due to its weak assumptions. The idea of operating VB in the collapsed space was proposed by Teh et al. (2007) and Sung et al. (2008), and it was successfully applied to "bag-of-words" models such as latent Dirichlet allocation (LDA) (Teh et al., 2007) and mixture of Gaussian (Sung et al., 2008), where the latent variables are conditionally independent given the parameters. By combining the CVB idea and the dynamic programming techniques used in structurally dependent models, we deliver a both efficient and accurate algorithm for training PCFGs and other structured natural language models.

The rest of the paper is structured as follows. We begin with the Bayesian models of PCFGs, and relate the existing training algorithms. Section 3 introduces collapsed variational Bayesian inference for "bag-of-words" models (defined in Section 3.1). We discuss the difficulty in applying such inference to structured models, followed by an approximate CVB algorithm for PCFGs.

*Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 173–182,
Sofia, Bulgaria, August 8-9 2013. ©2013 Association for Computational Linguistics

An alternative approach is also included in brief. In Section 4, we validate our CVB algorithm in three simple experiments. They are inferring a sparse grammar that describes the morphology of the Sotho language (Johnson et al., 2007), unsupervised dependency parsing (Klein and Manning, 2004) and supervised parsing with latent annotations (Matsuzaki et al., 2005). Section 5 concludes with future work.

## 2 Approximate inference for PCFGs

### 2.1 Definitions

A PCFG is a tuple $(T, N, S, R, \theta)$, where $T$, $N$, $R$ and $\theta$ are the finite sets of terminals, nonterminals, rules and parameters respectively, and $S \in N$ is the start symbol. We adopt a similar notation to Johnson et al. (2007), and assume that the context free grammar $G = (T, N, S, R)$ is in Chomsky normal form and the empty string $\epsilon \notin T$. Hence, each rule $r \in R$ takes either the form $A \to BC$ or $A \to w$, where $A, B, C \in N$ and $w \in T$. Let $\theta_{A\to\beta}$ be the probability of derivation rule $A \to \beta$, where $\beta$ ranges over $(N \times N) \cup T$. In the Bayesian setting, we place Dirichlet priors with hyperparameters $\alpha_A = \{\alpha_{A\to\beta}\}$ on each $\theta_A$ $= \{\theta_{A\to\beta}\}$.

Given a corpus of sentences $\mathbf{w} = (w_1, ..., w_n)$ and the corresponding hidden parse trees $\mathbf{t} = (t_1, ..., t_n)$, the joint probability distribution of parameters and variables is[1]:

$$\begin{aligned} P(\mathbf{w}, \mathbf{t}, \theta|\alpha) =& P(\theta|\alpha) \prod_{i=1}^{n} P_G(w_i, t_i|\theta) \\ =& \Big( \prod_{A \in N} P_D(\theta_A|\alpha_A) \Big) \prod_{r \in R} \theta_r^{f_r(\mathbf{t})} \end{aligned} \tag{1}$$

$$\begin{aligned} P_D(\theta_A|\alpha_A) =& \frac{1}{B(\alpha_A)} \prod_{r \in R_A} \theta_r^{\alpha_r - 1} \\ B(\alpha_A) =& \frac{\prod_{r \in R_A} \Gamma(\alpha_r)}{\Gamma(\sum_{r \in R_A} \alpha_r)} \end{aligned}$$

where $f_r(\mathbf{t})$ is the frequency of product rule $r$ in all the parse trees $\mathbf{t}$, and $R_A$ is the set of rules with left-hand side $A$. For a Dirichlet distribution $P_D(\theta_A|\alpha_A)$, $B(\alpha_A)$ is the normalization constant that can be written in terms of the gamma function $\Gamma$ (i.e. the generalised factorial function).

### 2.2 Variational Bayesian inference

The standard inside-outside algorithm for PCFGs belongs to the general EM class, which is further a subclass of VB (Beal, 2003). VB maximises the negative free energy $-\mathcal{F}(Q(\mathbf{t}, \theta))$, a lower bound of the log marginal likelihood of the observation $\log P(\mathbf{w}|\alpha)$. This is equivalent to minimising the Kullback-Leibler divergence.

$$\begin{aligned} &\log P(\mathbf{w}|\alpha) \geq -\mathcal{F}(Q(\mathbf{t}, \theta)) \\ =& \mathbb{E}_{Q(\mathbf{t},\theta)}[\log P(\mathbf{w}, \mathbf{t}, \theta|\alpha)] - \mathbb{E}_{Q(\mathbf{t},\theta)}[\log Q(\mathbf{t}, \theta)] \end{aligned}$$

$Q(\mathbf{t}, \theta)$ is an approximate posterior, where the parameters and hidden variables are assumed to be independent. Thus, it is factorised:

$$Q(\mathbf{t}, \theta) \approx Q(\mathbf{t})Q(\theta) \tag{2}$$

This strong independence assumption allows for the separate updates of $Q(\mathbf{t})$ and $Q(\theta)$ iteratively, optimising the negative free energy $-\mathcal{F}(Q(\mathbf{z}, \theta))$. For the traditional IO algorithm using maximum likelihood estimation, $Q(\theta)$ is further assumed to be degenerate, i.e. $Q(\theta) = \delta(\theta = \theta^\star)$.

$$\begin{aligned} &\text{E step: } Q(\mathbf{t}) \propto \exp(\mathbb{E}_{Q(\theta)}[\log P(\mathbf{w}, \mathbf{t}, \theta)]) \\ &\text{M step: } \theta^\star = \operatorname*{argmax}_{\theta} P(\mathbf{w}, \mathbf{t}, \theta) \end{aligned}$$

In the E step, we update $Q(\mathbf{t})$. For each tree $t$,

$$\begin{aligned} Q(t) &\propto P_G(w, t|\theta^\star) \\ &= \prod_{r \in R} (\theta_r^\star)^{f_r(t)} \end{aligned} \tag{3}$$

The distribution over parse tree $Q(t)$ is intractable to compute as its normalization requires summing over all possible parse trees producing $w$. We use dynamic programming to compute inside and outside probabilities recursively with the aim of accumulating the expected counts.

$$\begin{aligned} \mathbb{E}[f_{A\to BC}(t)|w] \propto& \sum_{0 \leq i < j < k \leq |w|} P_{\text{OUT}}(A, i, k) \times \\ & \theta_{A\to BC} P_{\text{IN}}(B, i, j) P_{\text{IN}}(C, j, k) \\ \mathbb{E}[f_{A\to w}(t)|w] \propto& \sum_{0 \leq i \leq |w|} P_{\text{OUT}}(A, i) \times \\ & \theta_{A\to w_i} \delta(w_i = w) \end{aligned}$$

where $P_{\text{IN}}(A, i, k)$ is the inside probability of observation $w_{i,k} = w_i, ..., w_k$ given $A$ is the root of the subtree, and $P_{\text{OUT}}(A, i, k)$ is the probability of $A$ spanning $(i, k)$, together with the rest of $w$.

[1] Strictly speaking, for each $(w, t)$ pair, if a hidden tree $t$ is arbitrary, we need to include two delta functions, namely $\delta(w = \text{yield}(t))$ and $\delta(G \Rightarrow^\star t)$. We assume that both delta functions are true, otherwise the probability of such pair is 0.

In the M step, we find the optimal $\theta^\star$ based on the MLE principle:

$$\theta^\star_{A\to\beta} = \frac{\mathbb{E}[f_{A\to\beta}(\mathbf{t})|\mathbf{w}]}{\sum_{A\to\beta'\in R_A}\mathbb{E}[f_{A\to\beta'}(\mathbf{t})|\mathbf{w}]}$$

$$\mathbb{E}[f_{A\to\beta}(\mathbf{t})|\mathbf{w}] = \sum_{i=1}^{n}\mathbb{E}[f_{A\to\beta}(t_i)|w_i]$$

VB inference is the generalisation of EM in the sense that it allows arbitrary parametric forms of $Q(\theta)$. Thus, the update equation in the M step is:

$$Q(\theta) \propto \exp(\mathbb{E}_{Q(\mathbf{t})}[\log P(\mathbf{w}, \mathbf{t}, \theta|\alpha)])$$

By the conjugacy property, the new $Q(\theta)$ is still in Dirichlet distribution form except with updated hyperparameters as shown by Kurihara and Sato (2006). Instead, Beal (2003) suggested an equivalent mean parameters $\tilde{\theta}$. Based on implementation of the EM algorithm, we only need a minor modification in the M step.

$$\tilde{\theta}_{A\to\beta} = \frac{m(\mathbb{E}[f_{A\to\beta}(\mathbf{t})|\mathbf{w}]+\alpha_{A\to\beta})}{m(\sum_{A\to\beta'\in R_A}(\mathbb{E}[f_{A\to\beta'}(\mathbf{t})|\mathbf{w}]+\alpha_{A\to\beta'}))}$$

$$m(x) = \exp(\Psi(x))$$

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

From the joint distribution in (1) proportional to the true posterior, we notice that the parameters and hidden variables are intimately coupled. Fluctuations in the parameters can induce changes in the hidden variables and vice-versa. Hence, the independence assumption in (2) and Figure 1(d) seems too strong, leading to inaccurate local maximums, although it allows for efficient and deterministic updates in EM and VB. The dependencies between parameters and hidden variables are kept intact for the remaining algorithms in this paper.

### 2.3 Markov Chain Monte Carlo

The standard Gibbs sampler for PCFGs iteratively samples the parameters $\theta$ and all the parse trees $\mathbf{t}$. Its mixing can be slowed by again the strong dependencies between the parameters and hidden variables. Instead of reparsing all the hidden trees $\mathbf{t}$ for each sample of $\theta$, collapsed Gibbs sampling (CGS) improves upon Gibbs sampling in terms of convergence speed by integrating out the parameters, and sampling directly from $P(\mathbf{t}|\mathbf{w}, \alpha)$ in a component-wise manner. Thus, it also deals with the dependencies exactly.

By using the conjugacy property, we can easily compute the marginal distribution of $\mathbf{w}$ and $\mathbf{t}$:

$$\begin{aligned} P(\mathbf{w}, \mathbf{t}|\alpha) &= \int_\theta P_G(\mathbf{w}, \mathbf{t}|\theta)P_D(\theta|\alpha)d\theta \\ &= \prod_{A\in N}\frac{B(\mathbf{f}_A(\mathbf{t}) + \alpha_A)}{B(\alpha_A)} \end{aligned} \quad (4)$$

where we define $\mathbf{f}_A(\mathbf{t})$ to be a vector of rule frequencies in $\mathbf{t}$ indexed by $A \to \beta \in R_A$. Hence, the conditional distribution for a parse tree $t_i$ given all others is:

$$\begin{aligned} P(t_i|w_i, \mathbf{w}^{\neg i}, \mathbf{t}^{\neg i}, \alpha) &\propto P(w_i, t_i|\mathbf{w}^{\neg i}, \mathbf{t}^{\neg i}, \alpha) \\ &= \prod_{A\in N}\frac{B(\mathbf{f}_A(\mathbf{t}) + \alpha_A)}{B(\mathbf{f}_A(\mathbf{t}^{\neg i}) + \alpha_A)} \end{aligned} \quad (5)$$

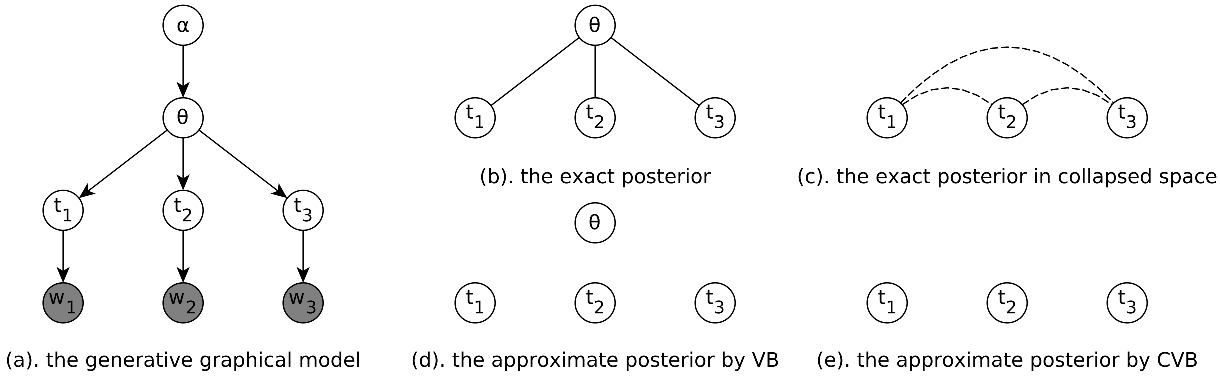
where $\mathbf{w}^{\neg i}$ and $\mathbf{t}^{\neg i}$ denote all other sentences and trees. It is noticeable that sampling a parse tree from the above conditional distribution is difficult. The frequencies $\mathbf{f}_A(\mathbf{t})$ effectively mean that the production probabilities are dependent on the current parse tree $t_i$. That is rule parameters can be updated on the fly inside a parse tree, which prohibits efficient dynamic programming tricks.

In order to solve this problem, Johnson et al. (2007) proposed a Hastings sampler that specified an alternative rule probabilities $\theta^{\mathrm{H}}$ of a proposal distribution $P(t_i|w_i, \theta^{\mathrm{H}})$, where

$$\theta^{\mathrm{H}}_{A\to\beta} = \frac{f_{A\to\beta}(\mathbf{t}^{\neg i}) + \alpha_{A\to\beta}}{\sum_{A\to\beta'\in R_A}(f_{A\to\beta'}(\mathbf{t}^{\neg i}) + \alpha_{A\to\beta'})}$$

The rule probabilities $\theta^{\mathrm{H}}$ are based on the statistics collected from all other parse trees, and they are fixed for the conditional distribution of the current parse tree. Therefore, by using a variant of inside algorithm (Goodman, 1998), one can efficiently sample a parse tree, which will be either accepted or rejected based on the Metropolis choice.

The MCMC based algorithms do not make any assumptions at all, and they can converge to the true posterior, either in joint or collapsed space as shown in Figure 1(b), 1(c). However, one needs to have experience about the number of samples to be collected and the burn-in period. For computationally intensive tasks such as learning PCFGs from a large corpus, a sufficiently large number of samples are required to decrease the sampling variance. Therefore, MCMC algorithms improves the performance over EM and VB at the cost of much more training time.



(a). the generative graphical model

(b). the exact posterior

(c). the exact posterior in collapsed space

(d). the approximate posterior by VB

(e). the approximate posterior by CVB

Figure 1: Graphical representations of the PCFG with $n = 3$ trees (a), and the (approximate) posteriors for Gibbs sampling (b), collapsed Gibbs sampling (c), variational Bayesian inference (d), and collapsed variational Bayesian inference (e). We use dashed lines to depict the weak dependencies.

## 3 Collapsed variational Bayesian inference

### 3.1 For bag-of-words models

Leveraging the insight that a sampling algorithm in collapsed space mixes faster than the standard one, Teh et al. (2007) proposed a similar argument that a VB inference algorithm in collapsed space is more effective than the standard one. Following the success in LDA (Teh et al., 2007), a number of research results have been accumulated around applying CVB to a variety of "bag-of-words" models (Sung et al., 2008; Sato et al., 2012; Wang and Blei, 2012).

Formally, we define a model to be independent and identically distributed (i.i.d.) (or informally "bag-of-words") if its hidden variables are conditionally independent given the parameters. LDA, IBM word alignment model 1 and 2, and various finite mixture models are typical examples.

For an i.i.d. model, integrating out parameters induces dependencies that spread over many hidden variables, and thus the dependency between any two variables is very weak. This provides an ideal setting to apply the mean field method (i.e. fully factorized VB), as its underlying assumption is that any variable depends on only the summary statistics collected from other variables called the field, and any particular variable's impact on the field is very small. Hence, the mean field assumption is better satisfied in collapsed space with very weak dependencies than in joint space with strong dependencies. As a result, we expect that VB in collapsed space can achieve more accurate results than the standard VB, and the results would be very close to the true posterior.

Even in collapsed space, CVB remains a deterministic algorithm that updates the posterior distributions over the hidden variables just like VB and EM. Therefore, we expect CVB to be computationally efficient as well.

### 3.2 For structured NLP models

We notice that the basic condition for applying the CVB algorithm to a specific model is for the model to be i.i.d., such that the hidden variables are only weakly dependent in collapsed space, providing an ideal condition to operate VB. However, the i.i.d. condition is certainly not true for structured NLP models such as hidden Markov models (HMMs) and PCFGs. Given the shape of a parse tree, a hidden variable is strongly dependent on its parent, siblings and children, and weakly dependent on the rest. Even worse, to infer a grammar from terminal strings, we don't even have access to the shape of parse trees, let alone analyzing the dependencies of hidden variables inside trees.

Although the PCFG model is not i.i.d. at the variable level, we can lift the idea of CVB up to the tree level. As our research domain is those large scale applications in language processing, a common feature of those problems is that there are usually many sentences, each of which has a hidden parse tree behind it. Hence, we may consider each sentence together with its parse tree to be drawn i.i.d. from the same set of parameters. Therefore, at the tree level, a PCFG can be considered as an i.i.d. model as shown in Figure 1(a) and thus, it can be fitted in the CVB framework as described in Section 3.1. We summarise the as-

$$Q(t_i) \propto \prod_{A \in N} \frac{\prod_{A\to\beta\in R_A} \exp(\mathbb{E}_{Q(\mathbf{t}^{\neg i})}[\log(\prod_{j=0}^{f_{A\to\beta}(t_i)-1}(f_{A\to\beta}(\mathbf{t}^{\neg i}) + \alpha_{A\to\beta} + j))])}{\prod_{j=0}^{(\sum_{A\to\beta'} f_{A\to\beta'}(t_i))-1} \exp(\mathbb{E}_{Q(\mathbf{t}^{\neg i})}[\log(\sum_{A\to\beta'\in R_A}(f_{A\to\beta}(\mathbf{t}^{\neg i}) + \alpha_{A\to\beta'} + j))])}$$

Figure 2: The exact mean field update in collapsed space for the parse tree $t_i$.

$$Q(t_i) \propto \prod_{r=A\to\beta\in R} \left( \frac{\mathbb{E}_{Q(\mathbf{t}^{\neg i})}[f_{A\to\beta}(\mathbf{t}^{\neg i})] + \alpha_{A\to\beta}}{\sum_{A\to\beta'}(\mathbb{E}_{Q(\mathbf{t}^{\neg i})}[f_{A\to\beta'}(\mathbf{t}^{\neg i})] + \alpha_{A\to\beta'})} \right)^{f_r(t_i)}$$

Figure 3: The approximate mean field update in collapsed space for the parse tree $t_i$.

sumptions made by each algorithm in Figure 1(b-e) before presenting the CVB algorithm formally.

The CVB algorithm for the PCFG model keeps the dependencies between the parameters and the hidden parse trees in an exact fashion:

$$Q(\mathbf{t}, \theta) = Q(\mathbf{t})Q(\theta|\mathbf{t})$$

We factorise $Q(\mathbf{t})$ by breaking only the weak dependencies between parse trees, while keeping the inside dependencies intact, as we don't make further assumptions about $Q(t)$ for each $t$.

$$Q(\mathbf{t}) \approx \prod_{i=1}^{n} Q(t_i)$$

By the above factorisations, we compute the negative variational free energy $-\mathcal{F}(Q(\mathbf{t})Q(\theta|\mathbf{t}))$ as follows:

$$\begin{aligned} & -\mathcal{F}(Q(\mathbf{t})Q(\theta|\mathbf{t})) \\ =& \mathbb{E}_{Q(\mathbf{t})Q(\theta|\mathbf{t})}[\log P(\mathbf{w}, \mathbf{t}, \theta|\alpha) - \log Q(\mathbf{t})Q(\theta|\mathbf{t})] \\ =& \mathbb{E}_{Q(\mathbf{t})}[\mathbb{E}_{Q(\theta|\mathbf{t})}[\log \frac{P(\mathbf{w}, \mathbf{t}, \theta|\alpha)}{Q(\theta|\mathbf{t})}] - \log Q(\mathbf{t})] \end{aligned}$$

Maximizing $-\mathcal{F}(Q(\mathbf{t})Q(\theta|\mathbf{t}))$ requires to update $Q(\theta|\mathbf{t})$ and $Q(\mathbf{t})$ in turn. In particular, $Q(\theta|\mathbf{t})$ is set equal to the true posterior $P(\theta|\mathbf{w}, \mathbf{t}, \alpha)$:

$$\begin{aligned} & -\mathcal{F}(Q(\mathbf{t})P(\theta|\mathbf{w}, \mathbf{t})) \\ =& \mathbb{E}_{Q(\mathbf{t})}[\mathbb{E}_{P(\theta|\mathbf{w},\mathbf{t},\alpha)}[\log \frac{P(\mathbf{w}, \mathbf{t}, \theta|\alpha)}{P(\theta|\mathbf{w}, \mathbf{t}, \alpha)}] - \log Q(\mathbf{t})] \\ =& \mathbb{E}_{Q(\mathbf{t})}[\log P(\mathbf{w}, \mathbf{t}|\alpha) - \log Q(\mathbf{t})] \end{aligned}$$

Finally, we update the approximate posterior for each parse tree $t$ by using the mean field method in the collapsed space:

$$Q(t_i) \propto \exp(\mathbb{E}_{Q(\mathbf{t}^{\neg i})}[\log P(w_i, t_i|\mathbf{w}^{\neg i}, \mathbf{t}^{\neg i}, \alpha)]) \tag{6}$$

The inner term $P(w_i, t_i|\mathbf{w}^{\neg i}, \mathbf{t}^{\neg i}, \alpha)$ in the above equation is just the unnormalized collapsed Gibbs sampling in (5). Plugging in (5), and expanding terms such as $B(\alpha_A)$ and $\Gamma(x)$, we obtain an exact computation of $Q(t_i)$ in Figure 2.

The exact computation is both intractable and expensive. The intractability comes from the similar problem as in the collapsed Gibbs sampling that we are unable to calculate the normalisation term $\sum_{t_i} Q(t_i)$. Hence, we follow Johnson et al. (2007) to approximate it by using only the statistics from other sentences, namely $\theta^{\mathrm{H}}$ and ignoring the local contribution.

$$P(w_i, t_i|\mathbf{w}^{\neg i}, \mathbf{t}^{\neg i}, \alpha) \approx \prod_{A\to\beta\in R} (\theta^{\mathrm{H}}_{A\to\beta})^{f_{A\to\beta}(t_i)} \tag{7}$$

We discuss the accuracy of (7) in Section 3.3. For those expensive computations of the expected log counts in Figure 2, Teh et al. (2007) and Sung et al. (2008) suggested the use of a linear Gaussian approximation based on the law of large numbers.

$$\begin{aligned} & \mathbb{E}_{Q(\mathbf{t}^{\neg i})}[\log(f_{A\to\beta}(\mathbf{t}^{\neg i}) + \alpha_{A\to\beta})] \\ \approx & \log(\mathbb{E}_{Q(\mathbf{t}^{\neg i})}[f_{A\to\beta}(\mathbf{t}^{\neg i})] + \alpha_{A\to\beta}) \end{aligned} \tag{8}$$

Substituting (7) into (6), and employing the linear approximation, we derive an approximate CVB algorithm as shown in Figure 3. In addition, its form is much more simplified and interpretable compared with the exact computation in Figure 2.

The surprising similarity between the approximate CVB update in Figure 3 and E step update in (3) indicates that the dynamic programming used in both EM and VB can take over from now. To run inside-outside recursion, the EM algorithm employs the parameters $\theta^{\star}$ based on maximum likelihood estimation; the VB algorithm employs the mean parameters $\tilde{\theta}$; and our CVB algorithm employs the parameters $\theta^{\mathrm{CVB}}$ computed from the expected counts of all other sentences.

The implementation can be easily achieved by modifying code of the EM algorithm. We keep track of the expected counts at global level, subtract the local mean counts for $t_i$ before update, run the inside-outside recursion using $\theta^{\mathrm{CVB}}$, and finally add the updated distribution back into the global counts. Therefore, we only need to replace the parameters with the expected counts, and make update after each sentence; the core of the inside-outside implementation remains the same.

Our CVB algorithm bears some similarities to the online EM algorithm with maximum a posterior (MAP) updates (Neal and Hinton, 1998; Liang and Klein, 2009), but they differ in several ways. The online EM algorithm updates each tree $t_i$ based on the statistics of all the trees, optimising the same objective function $p(\mathbf{w}|\theta)$ as the batch EM algorithm. MAP estimation searches for the optimal posterior $p(\mathbf{w}|\theta)p(\theta)$. On the other hand, our CVB algorithm optimises the data likelihood $p(\mathbf{w})$. The smoothing effects for the MAP estimation ($\alpha_{A\to\beta} - 1$) prevent the use of sparse priors, whereas the CVB algorithm ($\alpha_{A\to\beta}$) overcomes such difficulty by parameter integration.

### 3.3 Discussion

Breaking the weak dependencies between hidden variables and employing the linear approximation have been argued to be accurate (Teh et al., 2007; Sung et al., 2008; Sato and Nakagawa, 2012), and they are the standard procedures in applying the CVB algorithms to i.i.d. models.

In our CVB algorithm for PCFGs, we introduce an extra approximation in (7), which we argue is accurate. Theoretically, the inaccuracy only occurs when there are repeated rules in a parse tree as shown in Figure 2, so the same rule seen later uses a slightly different probability. Even if the inaccuracy indeed occurs, in our described scenario of many sentences, the local contribution from a single sentence is small compared with the statistics from all other sentences. Empirically, we replicate the experiment of Setho language by Johnson et al. (2007) in Section 4.1, and we find that the sampled trees based on $\theta^{\mathrm{H}}$ never get rejected, illustrating an acceptance rate close to $100\%$, and meaning that $\theta^{\mathrm{H}}$ is a very accurate Metropolis proposal. Since all the assumptions made by the CVB algorithm are reasonable and weak, we expect its results to be close to true posteriors.

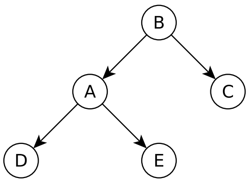


Figure 4: A fragment of a tree structure

### 3.4 An alternative approach

We briefly sketch an alternative CVB algorithm at the variable level for completeness.

For a structured NLP model with its shape to be fixed such as the PCFG with latent annotations (PCFG-LA) (Matsuzaki et al., 2005) (See definition in Section 4.3), we can simply ignore all the dependencies between the hidden variables in the collapsed space, despite whether they are strong (for adjacent nodes) or weak (for others). Although it seems that we have made unreasonable assumptions, it is not transparent which is worse comparing with the assumptions in the standard VB. Following this assumption, we can derive a CVB algorithm similar to the corresponding local sampling algorithm that samples one hidden variable at a time. For example, the approximate posterior over the subtype of the node $A$ in the above tree fragment in Figure 4 is updated follows:

$$\begin{aligned} & q(A = a) \\ \propto & \frac{\mathbb{E}[f_{B\to aC}(\mathbf{t}^{\neg A})] + \alpha}{\mathbb{E}[f_B(\mathbf{t}^{\neg A})] + |R_B|\alpha} \cdot \frac{\mathbb{E}[f_{a\to DE}(\mathbf{t}^{\neg A})] + \alpha}{\mathbb{E}[f_a(\mathbf{t}^{\neg A})] + |R_a|\alpha} \end{aligned}$$

where we use $A$ to denote the node position, and $a$ to denote its hidden subtype. $q(A = a)$ means the probability of node $A$ being in subtype $a$. In addition, we need to take into account the distributions over its adjacent variables. In our case, $A$ is strongly dependent on nodes $B, C, D, E$, and only weakly dependent on other variables (not shown in the above tree fragment) via global counts, e.g.:

$$\begin{aligned} & \mathbb{E}[f_{B\to aC}(\mathbf{t}^{\neg A})] \\ = & \sum_b \sum_c q(B = b)q(C = c)\mathbb{E}[f_{b\to ac}(\mathbf{t}^{\neg A})] \end{aligned}$$

However, it is not obvious how to use this alternative approach in general, and the performances of resulting algorithms remain unclear. Therefore, we implement only the CVB algorithm at the tree level in Section 3.2 for our experiments.

## 4 Experiments

We conduct three simple experiments to validate our CVB algorithm for PCFGs. In Section 4.1, we illustrate the significantly reduced training time of our CVB algorithm compared to the related Hastings algorithm; whereas in later two sections, we demonstrate the increased performance of our CVB algorithm compared to the corresponding VB and EM algorithms.

### 4.1 Inferring sparse grammars

Firstly, we conduct the same experiment of inferring sparse grammars describing the morphology of the Sotho language as in Johnson et al. (2007). We use the same corpus of unsegmented Sotho verb types from CHILDES (MacWhinney and Snow, 1985), and define the same initial CFG productions by allowing each non-terminal to emit any substrings in the corpus as terminals plus five predefined morphological rules at the top level.

We randomly withhold 10% of the verb types from the corpus for testing, and use the rest 90% for training. Both algorithms are evaluated by their per word perplexity on the test data set with prior set to $10^{-5}$ as suggested by Johnson et al. (2007). We run 5 times with random starts, and report the averaged results in Figure 5. The Hastings algorithm[2] takes roughly 1,000 iterations to converge, while our CVB algorithm reaches the convergence even before 10 iterations, consuming only a fraction of training time (CVB: 1.5 minutes; Hastings: 20 minutes). As well as little difference margin in final perplexities shown in Figure 5, we also evaluated segmentation quality measured by the F1 scores, and again the difference is trivial (CVB: 29.8%, Hastings: 31.3%).

### 4.2 Dependency model with valence

As a second empirical validation of our CVB inference algorithm, we apply it to unsupervised grammar induction with the popular Dependency Model with Valence (DMV) (Klein and Manning, 2004). Although the original maximum likelihood formulation of this model has long since been surpassed by more advanced models, all of the state-of-the-art approaches to unsupervised dependency parsing still have DMV at their core (Headden III et al., 2009; Blunsom and Cohn, 2010; Spitkovsky et al., 2012). As such we believe demonstrating improved inference on this core model will enable future improvements to more complex models.

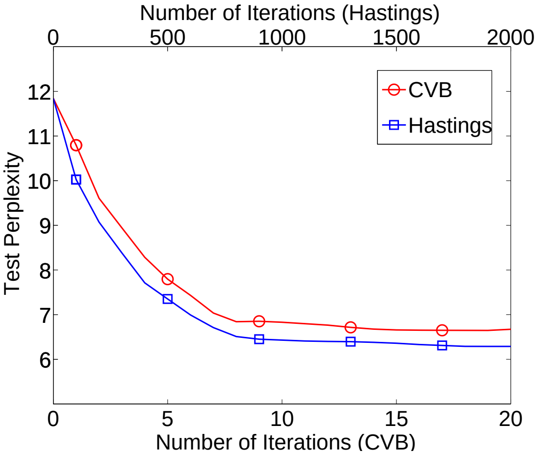


Figure 5: Perplexities averaged over 5 runs on the extracted corpus of Sotho verbs.

We evaluate a Dirichlet-Multinomial formulation of DMV in the standard fashion by training on sections 2-21 and testing on section 23 of the Penn. Wall Street Journal treebank (Marcus et al., 1993). We initialise our models using the original harmonic initialiser (Klein and Manning, 2004). Figure 6 displays the directed accuracy results for DMV model trained with CVB and VB with Dirichlet $\alpha$ parameters of either 1 or 0.1, as well as the previously reported MLE result. In both cases we see superior results for CVB inference, providing evidence that CVB may be a better choice of inference algorithm for Bayesian formulations of generative grammar induction models such as DMV.

### 4.3 PCFG with latent annotations

The vanilla PCFGs estimated by simply taking the empirical rule frequencies off treebanks are not accurate models to capture the syntactic structures in most natural languages as demonstrated by Charniak (1997) and Klein and Manning (2003). Our third experiment is to apply the CVB algorithm to the PCFGs with latent annotations (PCFGs-LA) (Matsuzaki et al., 2005), where each non-terminal symbol is augmented with hidden variables (or subtypes). Given a parsed corpus, training a PCFG-LA yields a finer grammar with the automatically induced features represented by the subtypes. For example, an augmented binary rule takes the form $A[a] \rightarrow B[b]C[c]$, where $a, b, c \in [1, H]$ are the hidden subtypes, and $H$ denotes the number of subtypes for each non-terminal.

[2] Annealing is not used in order to facilitate the perplexity calculation in the test set.

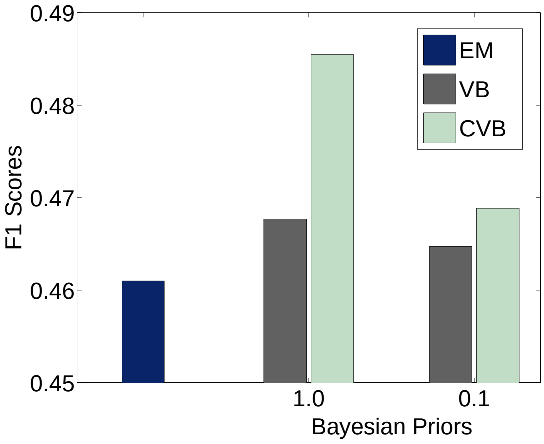


Figure 6: DMV trained by EM, VB and CVB. F1 scores on section 23, WSJ.

| Objective | Precision | Recall | F1 | Exact |
|---|---|---|---|---|
| EM | 75.84 | 72.92 | 74.35 | 11.13 |
| VB | 76.98 | 73.32 | 75.11 | 11.49 |
| CVB | 78.85 | 76.98 | 77.90 | 12.56 |

Table 1: PCFG-LA (2 subtypes) trained by EM, VB and CVB. Precision, Recall, F1 scores, Exact match scores on section 23, WSJ.

We follow the same experiment set-up as DMV, and report the results on the section 23, using the best grammar tested on the development set (section 22) from 5 random runs for each algorithm. We adopt Petrov et al. (2006)'s methods to process the data: right binarising and replacing infrequent words with the generic unknown word marker for English, and to initialise: adding 1% randomness to the parameters $\theta_0$ to start the EM training. We calculate the expected counts from $(G, \theta_0)$ to initialise our VB and CVB algorithms.

In Table 1, when each non-terminal is split into 2 hidden subtypes, we show that our CVB algorithm outperforms the EM and VB algorithms in terms of all the evaluation objectives. We also investigate the hidden state space with higher dimensions (4,8,16 subtypes), and find our CVB algorithm retains the advantages over the other two, whereas the VB algorithm fails to surpass the EM algorithm as reported in Figure 7.

## 5 Conclusion and future work

In this paper we have presented a collapsed variational Bayesian inference algorithm for PCFGs. We make use of the common scenario where the data consists of multiple short sentences, such that

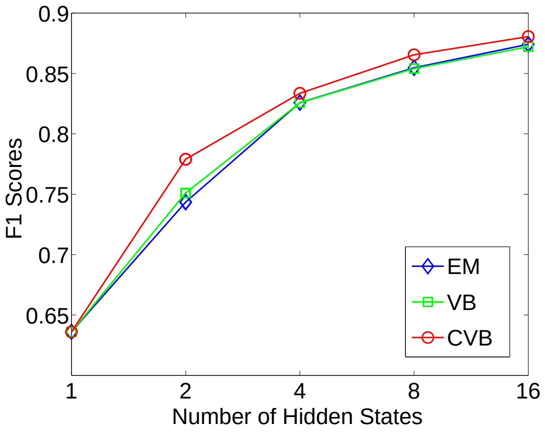


Figure 7: PCFG-LA (2,4,8,16 subtypes) trained by EM, VB and CVB. F1 scores on section 23, WSJ.

we can ignore the local dependencies induced by collapsing the parameters. The assumptions in our CVB algorithm are reasonable for a range of parsing applications and justified in three tasks by the empirical observations: it produces more accurate results than standard VB, and close results to sampling with significantly less training time.

While not state-of-the-art, the models we have demonstrated our CVB algorithm on underlie a number of high performance grammar induction and parsing systems (Cohen and Smith, 2009; Blunsom and Cohn, 2010; Petrov and Klein, 2007; Liang et al., 2007). Therefore, our work naturally extends to employing our CVB algorithm in more advanced models such as hierarchical splitting and merging system used in Berkeley parser (Petrov and Klein, 2007), and generalising our CVB algorithm to the non-parametric models such as tree substitution grammars (Blunsom and Cohn, 2010) and infinite PCFGs (Liang et al., 2007).

We have also sketched an alternative CVB algorithm which makes a harsher independence assumption for the latent variables but then requires no approximation of the variational posterior by performing inference individually for each parse node. This model breaks some strong dependencies within parse trees, but if we expect the posterior to be highly skewed by using a sparse prior, the product of constituent marginals may well be a good approximation. We leave further exploration of this algorithm for future work.

## Acknowledgments

We would like to thank Mark Johnson for the data used in Section 4.1 and valuable advice.

## References

James K. Baker. 1979. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132.

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

Phil Blunsom and Trevor Cohn. 2010. Unsupervised induction of tree substitution grammars for dependency parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1204–1213, Cambridge, MA, October. Association for Computational Linguistics.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, AAAI'97/IAAI'97, pages 598–603. AAAI Press.

Shay B. Cohen and Noah A. Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 74–82, Morristown, NJ, USA. Association for Computational Linguistics.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

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

Joshua T. Goodman. 1998. *Parsing inside-out*. Ph.D. thesis, Cambridge, MA, USA. Adviser-Stuart Shieber.

William P. Headden III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 101–109, Boulder, Colorado, June.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2006. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In *NIPS*.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007. 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.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24:613–632.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 478.

Kenichi Kurihara and Taisuke Sato. 2006. Variational bayesian grammar induction for natural language. In *Proceedings of the 8th international conference on Grammatical Inference: algorithms and applications*, ICGI'06, pages 84–96, Berlin, Heidelberg. Springer-Verlag.

Percy Liang and Dan Klein. 2009. Online EM for unsupervised models. In *Proceedings HLT/NAACL*.

Percy Liang, Slav Petrov, Michael Jordan, and Dan Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In *Proc. of the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP-2007)*, pages 688–697, Prague, Czech Republic.

Brian MacWhinney and Catherine Snow. 1985. The child language data exchange system. *Child Language*.

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

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 75–82, Stroudsburg, PA, USA. Association for Computational Linguistics.

Radford Neal and Geoffrey E. Hinton. 1998. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational*

*Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 433–440, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Issei Sato, Kenichi Kurihara, and Hiroshi Nakagawa. 2012. Practical collapsed variational bayes inference for hierarchical dirichlet process. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 105–113, New York, NY, USA. ACM.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2012. Three dependency-and-boundary models for grammar induction. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*.

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

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

Chong Wang and David Blei. 2012. Truncation-free stochastic variational inference for bayesian nonparametric models. In *Neural Information Processing Systems*.