Shared Logistic Normal Distributions for Soft Parameter Tying in Unsupervised Grammar Induction

Shay B. Cohen and **Noah A. Smith**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{scohen,nasmith}@cs.cmu.edu

Abstract

We present a family of priors over probabilistic grammar weights, called the shared logistic normal distribution. This family extends the partitioned logistic normal distribution, enabling factored covariance between the probabilities of different derivation events in the probabilistic grammar, providing a new way to encode prior knowledge about an unknown grammar. We describe a variational EM algorithm for learning a probabilistic grammar based on this family of priors. We then experiment with unsupervised dependency grammar induction and show significant improvements using our model for both monolingual learning and *bilingual* learning with a non-parallel, multilingual corpus.

1 Introduction

Probabilistic grammars have become an important tool in natural language processing. They are most commonly used for parsing and linguistic analysis (Charniak and Johnson, 2005; Collins, 2003), but are now commonly seen in applications like machine translation (Wu, 1997) and question answering (Wang et al., 2007). An attractive property of probabilistic grammars is that they permit the use of well-understood parameter estimation methods for *learning*—both from labeled and unlabeled data. Here we tackle the unsupervised grammar learning problem, specifically for unlexicalized context-free dependency grammars, using an empirical Bayesian approach with a novel family of priors.

There has been an increased interest recently in employing Bayesian modeling for probabilistic grammars in different settings, ranging from putting priors over grammar probabilities (Johnson et al., 2007) to putting non-parametric priors over derivations (Johnson et al., 2006) to learning the set of states in a grammar (Finkel et al., 2007; Liang et al., 2007). Bayesian methods offer an elegant framework for combining prior knowledge with data. The main challenge in Bayesian grammar learning is efficiently approximating probabilistic inference, which is generally intractable. Most commonly variational (Johnson, 2007; Kurihara and Sato, 2006) or sampling techniques are applied (Johnson et al., 2006).

Because probabilistic grammars are built out of multinomial distributions, the Dirichlet family (or, more precisely, a collection of Dirichlets) is a natural candidate for probabilistic grammars because of its conjugacy to the multinomial family. Conjugacy implies a clean form for the posterior distribution over grammar probabilities (given the data and the prior), bestowing computational tractability.

Following work by Blei and Lafferty (2006) for topic models, Cohen et al. (2008) proposed an alternative to Dirichlet priors for probabilistic grammars, based on the logistic normal (LN) distribution over the probability simplex. Cohen et al. used this prior to softly tie grammar weights through the covariance parameters of the LN. The prior encodes information about which grammar rules' weights are likely to covary, a more intuitive and expressive representation of knowledge than offered by Dirichlet distributions.[1]

The contribution of this paper is two-fold. First, from the modeling perspective, we present a generalization of the LN prior of Cohen et al. (2008), showing how to extend the use of the LN prior to

[1] Although the task, underlying model, and weights being tied were different, Eisner (2002) also showed evidence for the efficacy of parameter tying in grammar learning.

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 74–82,
Boulder, Colorado, June 2009.

tie between *any* grammar weights in a probabilistic grammar (instead of only allowing weights within the same multinomial distribution to covary). Second, from the experimental perspective, we show how such flexibility in parameter tying can help in unsupervised grammar learning in the well-known monolingual setting and in a new *bilingual* setting where grammars for two languages are learned at once (without parallel corpora).

Our method is based on a distribution which we call the **shared logistic normal distribution**, which is a distribution over a collection of multinomials from different probability simplexes. We provide a variational EM algorithm for inference.

The rest of this paper is organized as follows. In §2, we give a brief explanation of probabilistic grammars and introduce some notation for the specific type of dependency grammar used in this paper, due to Klein and Manning (2004). In §3, we present our model and a variational inference algorithm for it. In §4, we report on experiments for both monolingual settings and a bilingual setting and discuss them. We discuss future work (§5) and conclude in §6.

2 Probabilistic Grammars and Dependency Grammar Induction

A probabilistic grammar defines a probability distribution over grammatical derivations generated through a step-by-step process. HMMs, for example, can be understood as a random walk through a probabilistic finite-state network, with an output symbol sampled at each state. Each "step" of the walk and each symbol emission corresponds to one derivation step. PCFGs generate phrase-structure trees by recursively rewriting nonterminal symbols as sequences of "child" symbols (each itself either a nonterminal symbol or a terminal symbol analogous to the emissions of an HMM). Each step or emission of an HMM and each rewriting operation of a PCFG is conditionally independent of the other rewriting operations given a single structural element (one HMM or PCFG state); this Markov property permits efficient inference for the probability distribution defined by the probabilistic grammar.

In general, a probabilistic grammar defines the joint probability of a string $\mathbf{x}$ and a grammatical derivation $\mathbf{y}$:

$$p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta}) = \prod_{k=1}^{K} \prod_{i=1}^{N_k} \theta_{k,i}^{f_{k,i}(\mathbf{x},\mathbf{y})} \quad (1)$$
$$= \exp \sum_{k=1}^{K} \sum_{i=1}^{N_k} f_{k,i}(\mathbf{x}, \mathbf{y}) \log \theta_{k,i}$$

where $f_{k,i}$ is a function that "counts" the number of times the kth distribution's ith event occurs in the derivation. The $\boldsymbol{\theta}$ are a collection of K multinomials $\langle \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K \rangle$, the kth of which includes N_k events. Note that there may be many derivations $\mathbf{y}$ for a given string $\mathbf{x}$—perhaps even infinitely many in some kinds of grammars.

2.1 Dependency Model with Valence

HMMs and PCFGs are the best-known probabilistic grammars, but there are many others. In this paper, we use the "dependency model with valence" (DMV), due to Klein and Manning (2004). DMV defines a probabilistic grammar for unlabeled, projective dependency structures. Klein and Manning (2004) achieved their best results with a combination of DMV with a model known as the "constituent-context model" (CCM). We do not experiment with CCM in this paper, because it does not fit directly in a Bayesian setting (it is highly deficient) and because state-of-the-art unsupervised dependency parsing results have been achieved with DMV alone (Smith, 2006).

Using the notation above, DMV defines $\mathbf{x} = \langle x_1, x_2, ..., x_n \rangle$ to be a sentence. x_0 is a special "wall" symbol, \$, on the left of every sentence. A tree $\mathbf{y}$ is defined by a pair of functions $\mathbf{y}_{left}$ and $\mathbf{y}_{right}$ (both $\{0, 1, 2, ..., n\} \rightarrow 2^{\{1,2,...,n\}}$) that map each word to its sets of left and right dependents, respectively. Here, the graph is constrained to be a *projective* tree rooted at $x_0 = \$$: each word except \$ has a single parent, and there are no cycles or crossing dependencies. $\mathbf{y}_{left}(0)$ is taken to be empty, and $\mathbf{y}_{right}(0)$ contains the sentence's single head. Let $\mathbf{y}^{(i)}$ denote the subtree rooted at position i. The probability $P(\mathbf{y}^{(i)} \mid x_i, \boldsymbol{\theta})$ of generating this subtree, given its head word x_i, is defined recursively, as described in Fig. 1 (Eq. 2).

The probability of the entire tree is given by $p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta}) = P(\mathbf{y}^{(0)} \mid \$, \boldsymbol{\theta})$. The $\boldsymbol{\theta}$ are the multinomial distributions $\theta_{\mathrm{s}}(\cdot \mid \cdot, \cdot, \cdot)$ and $\theta_{\mathrm{c}}(\cdot \mid \cdot, \cdot)$. To

$$P(\mathbf{y}^{(i)} \mid x_i, \boldsymbol{\theta}) = \prod_{D \in \{left, right\}} \theta_{\mathrm{s}}(\mathrm{stop} \mid x_i, D, [\mathbf{y}_D(i) = \emptyset]) \times \prod_{j \in \mathbf{y}_D(i)} \theta_{\mathrm{s}}(\neg\mathrm{stop} \mid x_i, D, \mathrm{first}_{\mathbf{y}}(j)) \times \theta_{\mathrm{c}}(x_j \mid x_i, D) \times P(\mathbf{y}^{(j)} \mid x_j, \boldsymbol{\theta}) \quad (2)$$

Figure 1: The "dependency model with valence" recursive equation. $\mathrm{first}_{\mathbf{y}}(j)$ is a predicate defined to be true iff x_j is the closest child (on either side) to its parent x_i. The probability of the tree $p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta}) = P(\mathbf{y}^{(0)} \mid \$, \boldsymbol{\theta})$.

follow the general setting of Eq. 1, we index these distributions as $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K$.

Headden et al. (2009) extended DMV so that the distributions θ_c condition on the valence as well, with smoothing, and showed significant improvements for short sentences. Our experiments found that these improvements do not hold on longer sentences. Here we experiment only with DMV, but note that our techniques are also applicable to richer probabilistic grammars like that of Headden et al.

2.2 Learning DMV

Klein and Manning (2004) learned the DMV probabilities $\boldsymbol{\theta}$ from a corpus of part-of-speech-tagged sentences using the EM algorithm. EM manipulates $\boldsymbol{\theta}$ to locally optimize the likelihood of the observed portion of the data (here, $\mathbf{x}$), marginalizing out the hidden portions (here, $\mathbf{y}$). The likelihood surface is not globally concave, so EM only locally optimizes the surface. Klein and Manning's initialization, though reasonable and language-independent, was an important factor in performance.

Various alternatives to EM were explored by Smith (2006), achieving substantially more accurate parsing models by altering the objective function. Smith's methods did require substantial hyperparameter tuning, and the best results were obtained using small annotated development sets to choose hyperparameters. In this paper, we consider only fully unsupervised methods, though we the Bayesian ideas explored here might be merged with the biasing approaches of Smith (2006) for further benefit.

3 Parameter Tying in the Bayesian Setting

As stated above, $\boldsymbol{\theta}$ comprises a collection of multinomials that weights the grammar. Taking the Bayesian approach, we wish to place a prior on those multinomials, and the Dirichlet family is a natural candidate for such a prior because of its conjugacy, which makes inference algorithms easier to derive. For example, if we make a "mean-field assumption," with respect to hidden structure and weights, the variational algorithm for approximately inferring the distribution over $\boldsymbol{\theta}$ and trees $\mathbf{y}$ resembles the traditional EM algorithm very closely (Johnson, 2007). In fact, variational inference in this case takes an action similar to smoothing the counts using the exp-Ψ function during the E-step. Variational inference can be embedded in an empirical Bayes setting, in which we optimize the variational bound with respect to the hyperparameters as well, repeating the process until convergence.

3.1 Logistic Normal Distributions

While Dirichlet priors over grammar probabilities make learning algorithms easy, they are limiting. In particular, as noted by Blei and Lafferty (2006), there is no explicit flexible way for the Dirichlet's parameters to encode beliefs about *covariance* between the probabilities of two events. To illustrate this point, we describe how a multinomial $\boldsymbol{\theta}$ of dimension d is generated from a Dirichlet distribution with parameters $\boldsymbol{\alpha} = \langle \alpha_1, ..., \alpha_d \rangle$:

1. Generate $\eta_j \sim \Gamma(\alpha_j, 1)$ independently for $j \in \{1, ..., d\}$.
2. $\theta_j \leftarrow \eta_j / \sum_i \eta_i$.

where $\Gamma(\alpha, 1)$ is a Gamma distribution with shape α and scale 1.

Correlation among θ_i and θ_j, $i \neq j$, cannot be modeled directly, only through the normalization in step 2. In contrast, LN distributions (Aitchison, 1986) provide a natural way to model such correlation. The LN draws a multinomial $\boldsymbol{\theta}$ as follows:

1. Generate $\boldsymbol{\eta} \sim \mathrm{Normal}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.
2. $\theta_j \leftarrow \exp(\eta_j) / \sum_i \exp(\eta_i)$.

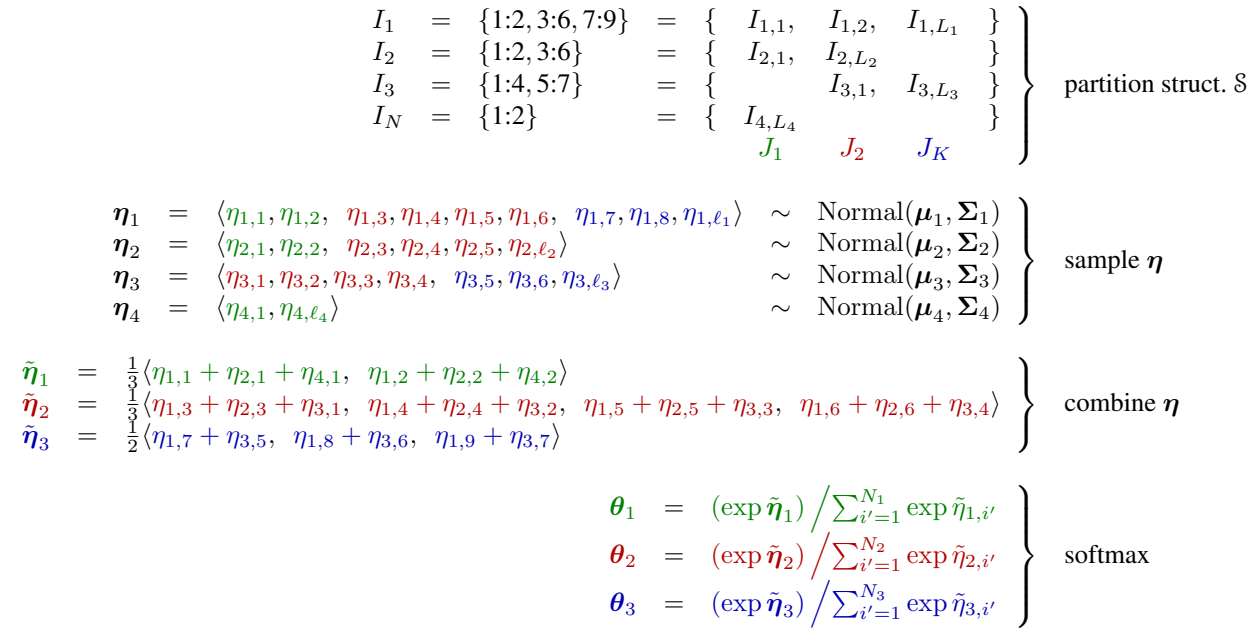

Figure 2: An example of a shared logistic normal distribution, illustrating Def. 1. $N = 4$ experts are used to sample $K = 3$ multinomials; $L_1 = 3$, $L_2 = 2$, $L_3 = 2$, $L_4 = 1$, $\ell_1 = 9$, $\ell_2 = 6$, $\ell_3 = 7$, $\ell_4 = 2$, $N_1 = 2$, $N_2 = 4$, and $N_3 = 3$. This figure is best viewed in color.

Blei and Lafferty (2006) defined correlated topic models by replacing the Dirichlet in latent Dirichlet allocation models (Blei et al., 2003) with a LN distribution. Cohen et al. (2008) compared Dirichlet and LN distributions for learning DMV using empirical Bayes, finding substantial improvements for English using the latter.

In that work, we obtained improvements even without specifying exactly *which* grammar probabilities covaried. While empirical Bayes learning permits these covariances to be discovered without supervision, we found that by initializing the covariance to encode beliefs about which grammar probabilities should covary, further improvements were possible. Specifically, we grouped the Penn Treebank part-of-speech tags into coarse groups based on the treebank annotation guidelines and biased the initial covariance matrix for each child distribution $\theta_c(\cdot \mid \cdot, \cdot)$ so that the probabilities of child tags from the same coarse group covaried. For example, the probability that a past-tense verb (VBD) has a singular noun (NN) as a right child may be correlated with the probability that it has a *plural* noun (NNS) as a right child. Hence linguistic knowledge—specifically, a coarse grouping of word classes—can be encoded in the prior.

A per-distribution LN distribution only permits probabilities within a multinomial to covary. We will generalize the LN to permit covariance among any probabilities in $\boldsymbol{\theta}$, throughout the model. For example, the probability of a past-tense verb (VBD) having a noun as a right child might correlate with the probability that other kinds of verbs (VBZ, VBN, etc.) have a noun as a right child.

The *partitioned logistic normal distribution* (PLN) is a generalization of the LN distribution that takes the first step towards our goal (Aitchison, 1986). Generating from PLN involves drawing a random vector from a multivariate normal distribution, but the logistic transformation is applied to different parts of the vector, leading to sampled multinomial distributions of the required lengths from different probability simplices. This is in principle what is required for arbitrary covariance between grammar probabilities, except that DMV has $O(t^2)$ weights for a part-of-speech vocabulary of size t, requiring a very large multivariate normal distribution with $O(t^4)$ covariance parameters.

3.2 Shared Logistic Normal Distributions

To solve this problem, we suggest a refinement of the class of PLN distributions. Instead of using a single normal vector for all of the multinomials, we use several normal vectors, partition each one and then *recombine* parts which correspond to the same multinomial, as a mixture. Next, we apply the logisitic transformation on the mixed vectors (each of which is normally distributed as well). Fig. 2 gives an example of a non-trivial case of using a SLN distribution, where three multinomials are generated from four normal experts.

We now formalize this notion. For a natural number N, we denote by $1{:}N$ the set $\{1, ..., N\}$. For a vector in $v \in \mathbb{R}^N$ and a set $I \subseteq 1{:}N$, we denote by v_I to be the vector created from v by using the coordinates in I. Recall that K is the number of multinomials in the probabilistic grammar, and N_k is the number of events in the kth multinomial.

Definition 1. *We define a shared logistic normal distribution with N "experts" over a collection of K multinomial distributions. Let $\boldsymbol{\eta}_n \sim \mathrm{Normal}(\boldsymbol{\mu}_n, \Sigma_n)$ be a set of multivariate normal variables for $n \in 1{:}N$, where the length of $\boldsymbol{\eta}_n$ is denoted ℓ_n. Let $I_n = \{I_{n,j}\}_{j=1}^{L_n}$ be a partition of $1{:}\ell_n$ into L_n sets, such that $\cup_{j=1}^{L_n} I_{n,j} = 1{:}\ell_n$ and $I_{n,j} \cap I_{n,j'} = \emptyset$ for $j \neq j'$. Let J_k for $k \in 1{:}K$ be a collection of (disjoint) subsets of $\{I_{n,j} \mid n \in 1{:}N, j \in 1{:}\ell_n, |I_{n,j}| = N_k\}$, such that all sets in J_k are of the same size, N_k. Let $\tilde{\boldsymbol{\eta}}_k = \frac{1}{|J_k|} \sum_{I_{n,j} \in J_k} \boldsymbol{\eta}_{n, I_{n,j}}$, and $\theta_{k,i} = \exp(\tilde{\eta}_{k,i}) \,/ \sum_{i'} \exp(\tilde{\eta}_{k,i'})$. We then say $\boldsymbol{\theta}$ distributes according to the* shared logistic normal distribution *with* partition structure $\mathcal{S} = (\{I_n\}_{n=1}^N, \{J_k\}_{k=1}^K)$ *and* normal experts $\{(\boldsymbol{\mu}_n, \Sigma_n)\}_{n=1}^N$ *and denote it by* $\boldsymbol{\theta} \sim \mathrm{SLN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathcal{S})$.

The partitioned LN distribution in Aitchison (1986) can be formulated as a shared LN distribution where $N = 1$. The LN collection used by Cohen et al. (2008) is the special case where $N = K$, each $L_n = 1$, each $\ell_k = N_k$, and each $J_k = \{I_{k,1}\}$.

The covariance among arbitrary $\theta_{k,i}$ is not defined directly; it is implied by the definition of the normal experts $\boldsymbol{\eta}_{n, I_{n,j}}$, for each $I_{n,j} \in J_k$. We note that a SLN can be represented as a PLN by relying on the distributivity of the covariance operator, and merging all the partition structure into one (perhaps sparse) covariance matrix. However, if we are interested in keeping a factored structure on the covariance matrices which generate the grammar weights, we cannot represent every SLN as a PLN.

It is convenient to think of each $\eta_{i,j}$ as a weight associated with a unique event's probability, a certain outcome of a certain multinomial in the probabilistic grammar. By letting different $\eta_{i,j}$ covary with each other, we loosen the relationships among $\theta_{k,j}$ and permit the model—at least in principle—to learn patterns from the data. Def. 1 also implies that we multiply several multinomials together in a product-of-experts style (Hinton, 1999), because the exponential of a mixture of normals becomes a product of (unnormalized) probabilities.

Our extension to the model in Cohen et al. (2008) follows naturally after we have defined the shared LN distribution. The generative story for this model is as follows:

1. Generate $\boldsymbol{\theta} \sim \mathrm{SLN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathcal{S})$, where $\boldsymbol{\theta}$ is a collection of vectors $\boldsymbol{\theta}_k$, $k = 1, ..., K$.
2. Generate $\mathbf{x}$ and $\mathbf{y}$ from $p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta})$ (i.e., sample from the probabilistic grammar).

3.3 Inference

In this work, the partition structure $\mathcal{S}$ is *known*, the sentences $\mathbf{x}$ are *observed*, the trees $\mathbf{y}$ and the grammar weights $\boldsymbol{\theta}$ are *hidden*, and the parameters of the shared LN distribution $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are *learned*.[2]

Our inference algorithm aims to find the posterior over the grammar probabilities $\boldsymbol{\theta}$ and the hidden structures (grammar trees $\mathbf{y}$). To do that, we use variational approximation techniques (Jordan et al., 1999), which treat the problem of finding the posterior as an optimization problem aimed to find the best approximation $q(\boldsymbol{\theta}, \mathbf{y})$ of the posterior $p(\boldsymbol{\theta}, \mathbf{y} \mid \mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathcal{S})$. The posterior q needs to be constrained to be within a family of tractable and manageable distributions, yet rich enough to represent good approximations of the true posterior. "Best approximation" is defined as the KL divergence between $q(\boldsymbol{\theta}, \mathbf{y})$ and $p(\boldsymbol{\theta}, \mathbf{y} \mid \mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathcal{S})$.

Our variational inference algorithm uses a mean-field assumption: $q(\boldsymbol{\theta}, \mathbf{y}) = q(\boldsymbol{\theta})q(\mathbf{y})$. The distribution $q(\boldsymbol{\theta})$ is assumed to be a LN distribution with

[2] In future work, we might aim to learn $\mathcal{S}$.

$$\log p(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathcal{S}) \geq \underbrace{\left(\sum_{n=1}^{N} \mathbb{E}_q\left[\log p(\boldsymbol{\eta}_k \mid \boldsymbol{\mu}_k, \Sigma_k)\right]\right) + \left(\sum_{k=1}^{K}\sum_{i=1}^{N_k} \tilde{f}_{k,i}\tilde{\psi}_{k,i}\right) + H(q)}_{B} \quad (3)$$

$$\tilde{f}_{k,i} \triangleq \sum_{\mathbf{y}} q(\mathbf{y}) f_{k,i}(\mathbf{x}, \mathbf{y}) \quad (4)$$

$$\tilde{\psi}_{k,i} \triangleq \tilde{\mu}^C_{k,i} - \log \tilde{\zeta}_k + 1 - \frac{1}{\tilde{\zeta}_k}\sum_{i'=1}^{N_k} \exp\left(\tilde{\mu}^C_{k,i} + \frac{(\tilde{\sigma}^C_{k,i})^2}{2}\right) \quad (5)$$

$$\tilde{\mu}^C_k \triangleq \frac{1}{|J_k|}\sum_{I_{r,j} \in J_k} \tilde{\mu}_{r,I_{r,j}} \quad (6)$$

$$(\tilde{\sigma}^C_k)^2 \triangleq \frac{1}{|J_k|^2}\sum_{I_{r,j} \in J_k} \tilde{\sigma}^2_{r,I_{r,j}} \quad (7)$$

Figure 3: Variational inference bound. Eq. 3 is the bound itself, using notation defined in Eqs. 4–7 for clarity. Eq. 4 defines expected counts of the grammar events under the variational distribution $q(\mathbf{y})$, calculated using dynamic programming. Eq. 5 describes the weights for the weighted grammar defined by $q(\mathbf{y})$. Eq. 6 and Eq. 7 describe the mean and the variance, respectively, for the multivariate normal eventually used with the weighted grammar. These values are based on the parameterization of $q(\theta)$ by $\tilde{\mu}_{i,j}$ and $\tilde{\sigma}^2_{i,j}$. An additional set of variational parameters is $\tilde{\zeta}_k$, which helps resolve the non-conjugacy of the LN distribution through a first order Taylor approximation.

all off-diagonal covariances fixed at zero (i.e., the variational parameters consist of a single mean $\tilde{\mu}_{k,i}$ and a single variance $\tilde{\sigma}^2_{k,i}$ for each $\theta_{k,i}$). There is an additional variational parameter, $\tilde{\zeta}_k$ per multinomial, which is the result of an additional variational approximation because of the lack of conjugacy of the LN distribution to the multinomial distribution. The distribution $q(\mathbf{y})$ is assumed to be defined by a DMV with unnormalized probabilities $\tilde{\boldsymbol{\psi}}$.

Inference optimizes the bound B given in Fig. 3 (Eq. 3) with respect to the variational parameters. Our variational inference algorithm is derived similarly to that of Cohen et al. (2008). Because we wish to *learn* the values of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, we embed variational inference as the E step within a variational EM algorithm, shown schematically in Fig. 4. In our experiments, we use this variational EM algorithm on a training set, and then use the normal experts' means to get a point estimate for $\boldsymbol{\theta}$, the grammar weights. This is called **empirical Bayesian estimation**. Our approach differs from maximum *a posteriori* (MAP) estimation, since we re-estimate the parameters of the normal experts. Exact MAP estimation is probably not feasible; a variational algorithm like ours might be applied, though better performance is expected from adjusting the SLN to fit the data.

4 Experiments

Our experiments involve data from two treebanks: the *Wall Street Journal* Penn treebank (Marcus et al., 1993) and the Chinese treebank (Xue et al., 2004). In both cases, following standard practice, sentences were stripped of words and punctuation, leaving part-of-speech tags for the unsupervised induction of dependency structure. For English, we train on §2–21, tune on §22 (without using annotated data), and report final results on §23. For Chinese, we train on §1–270, use §301–1151 for development and report testing results on §271–300.[3]

To evaluate performance, we report the fraction of words whose predicted parent matches the gold standard corpus. This performance measure is also known as attachment accuracy. We considered two parsing methods after extracting a point estimate for the grammar: the most probable "Viterbi" parse ($\text{argmax}_{\mathbf{y}}\, p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$) and the minimum Bayes risk (MBR) parse ($\text{argmin}_{\mathbf{y}}\, \mathbb{E}_{p(\mathbf{y}' \mid \mathbf{x}, \boldsymbol{\theta})}[\ell(\mathbf{y}; \mathbf{x}, \mathbf{y}')]$) with dependency attachment error as the loss function (Goodman, 1996). Performance with MBR parsing is consistently higher than its Viterbi counterpart, so we report only performance with MBR parsing.

4.1 Nouns, Verbs, and Adjectives

In this paper, we use a few simple heuristics to decide which partition structure $\mathcal{S}$ to use. Our heuris-

[3] Unsupervised training for these datasets can be costly, and requires iteratively running a cubic-time inside-outside dynamic programming algorithm, so we follow Klein and Manning (2004) in restricting the training set to sentences of ten or fewer words in length. Short sentences are also less structurally ambiguous and may therefore be easier to learn from.

Input: initial parameters $\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$, partition structure $\mathbb{S}$, observed data $\mathbf{x}$, number of iterations T
Output: learned parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$
$t \leftarrow 1$;
while $t \leq T$ **do**
  E-step (for $\ell = 1, ..., M$*) do*: **repeat**
    optimize B w.r.t. $\tilde{\mu}_r^{\ell,(t)}, r = 1, ..., N$;
    optimize B w.r.t. $\tilde{\sigma}_r^{\ell,(t)}, r = 1, ..., N$;
    update $\tilde{\zeta}_r^{\ell,(t)}, r = 1, ..., N$;
    update $\tilde{\boldsymbol{\psi}}_r^{\ell,(t)}, r = 1, ..., N$;
    compute counts $\tilde{\mathbf{f}}_r^{\ell,(t)}, r = 1, ..., N$;
  until *convergence of* B ;
  M-step: optimize B w.r.t. $\boldsymbol{\mu}^{(t)}$ and $\boldsymbol{\Sigma}^{(t)}$;
  $t \leftarrow t + 1$;
end
return $\boldsymbol{\mu}^{(T)}$, $\boldsymbol{\Sigma}^{(T)}$

Figure 4: Main details of the variational inference EM algorithm with empirical Bayes estimation of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. B is the bound defined in Fig. 3 (Eq. 3). N is the number of normal experts for the SLN distribution defining the prior. M is the number of training examples. The full algorithm is given in Cohen and Smith (2009).

tics rely mainly on the centrality of content words: nouns, verbs, and adjectives. For example, in the English treebank, the most common attachment errors (with the LN prior from Cohen et al., 2008) happen with a noun (25.9%) or a verb (16.9%) parent. In the Chinese treebank, the most common attachment errors happen with noun (36.0%) and verb (21.2%) parents as well. The errors being governed by such attachments are the direct result of nouns and verbs being the most common parents in these data sets.

Following this observation, we compare four different settings in our experiments (all SLN settings include one normal expert for each multinomial on its own, equivalent to the regular LN setting from Cohen et al.):

- TIEV: We add normal experts that tie all probabilities corresponding to a verbal parent (*any* parent, using the coarse tags of Cohen et al., 2008). Let $\boldsymbol{V}$ be the set of part-of-speech tags which belong to the verb category. For each direction D (left or right), the set of multinomials of the form $\theta_{\mathrm{c}}(\cdot \mid v, D)$, for $v \in \boldsymbol{V}$, all share a normal expert. For each direction D and each boolean value B of the predicate $\mathrm{first}_{\mathbf{y}}(\cdot)$, the set of multinomials $\theta_{\mathrm{s}}(\cdot \mid x, D, v)$, for $v \in \boldsymbol{V}$ share a normal expert.
- TIEN: This is the same as TIEV, only for nominal parents.
- TIEV&N: Tie both verbs and nouns (in separate partitions). This is equivalent to taking the union of the partition structures of the above two settings.
- TIEA: This is the same as TIEV, only for adjectival parents.

Since inference for a model with parameter tying can be computationally intensive, we first run the inference algorithm without parameter tying, and then add parameter tying to the rest of the inference algorithm's execution until convergence.

Initialization is important for the inference algorithm, because the variational bound is a non-concave function. For the expected values of the normal experts, we use the initializer from Klein and Manning (2004). For the covariance matrices, we follow the setting in Cohen et al. (2008) in our experiments also described in §3.1. For each treebank, we divide the tags into twelve disjoint tag families.[4] The covariance matrices for all dependency distributions were initialized with 1 on the diagonal, 0.5 between tags which belong to the same family, and 0 otherwise. This initializer has been shown to be more successful than an identity covariance matrix.

4.2 Monolingual Experiments

We begin our experiments with a monolingual setting, where we learn grammars for English and Chinese (separately) using the settings described above.

The attachment accuracy for this set of experiments is described in Table 1. The baselines include right attachment (where each word is attached to the word to its right), MLE via EM (Klein and Manning, 2004), and empirical Bayes with Dirichlet and LN priors (Cohen et al., 2008). We also include a "ceiling" (DMV trained using supervised MLE from the training sentences' trees). For English, we see that tying nouns, verbs or adjectives improves performance compared to the LN baseline. Tying both nouns and verbs improves performance a bit more.

[4] These are simply coarser tags: adjective, adverb, conjunction, foreign word, interjection, noun, number, particle, preposition, pronoun, proper noun, verb.

		attachment acc. (%)		
		≤ 10	≤ 20	all
English	Attach-Right	38.4	33.4	31.7
	EM (K&M, 2004)	46.1	39.9	35.9
	Dirichlet	46.1	40.6	36.9
	LN (CG&S, 2008)	59.4	45.9	40.5
	SLN, TIEV	60.2	46.2	40.0
	SLN, TIEN	60.2	46.7	40.9
	SLN, TIEV&N	61.3	47.4	41.4
	SLN, TIEA	59.9	45.8	40.9
	Biling. SLN, TIEV	†61.6	47.6	41.7
	Biling. SLN, TIEN	†61.8	**48.1**	†42.1
	Biling. SLN, TIEV&N	**62.0**	†48.0	**42.2**
	Biling. SLN, TIEA	61.3	47.6	41.7
	Supervised MLE	*84.5*	*74.9*	*68.8*
Chinese	Attach-Right	34.9	34.6	34.6
	EM (K&M, 2004)	38.3	36.1	32.7
	Dirichlet	38.3	35.9	32.4
	LN	50.1	40.5	**35.8**
	SLN, TIEV	†51.9	**42.0**	**35.8**
	SLN, TIEN	43.0	38.4	33.7
	SLN, TIEV&N	45.0	39.2	34.2
	SLN, TIEA	47.4	40.4	35.2
	Biling. SLN, TIEV	†51.9	**42.0**	**35.8**
	Biling. SLN, TIEN	48.0	38.9	33.8
	Biling. SLN, TIEV&N	†51.5	†41.7	35.3
	Biling. SLN, TIEA	**52.0**	41.3	35.2
	Supervised MLE	*84.3*	*66.1*	*57.6*

Table 1: Attachment accuracy of different models, on test data from the Penn Treebank and the Chinese Treebank of varying levels of difficulty imposed through a length filter. Attach-Right attaches each word to the word on its right and the last word to \$. Bold marks best overall accuracy per length bound, and † marks figures that are not significantly worse (binomial sign test, $p < 0.05$).

4.3 Bilingual Experiments

Leveraging information from one language for the task of disambiguating another language has received considerable attention (Dagan, 1991; Smith and Smith, 2004; Snyder and Barzilay, 2008; Burkett and Klein, 2008). Usually such a setting requires a parallel corpus or other annotated data that ties between those two languages.[5]

Our bilingual experiments use the English and Chinese treebanks, which are not parallel corpora, to train parsers for both languages jointly. Sharing information between those two models is done by softly tying grammar weights in the two hidden grammars.

We first merge the models for English and Chinese by taking a union of the multinomial families of each and the corresponding prior parameters. We then add a normal expert that ties between the parts of speech in the respective partition structures for both grammars together. Parts of speech are matched through the single coarse tagset (footnote 4). For example, with TIEV, let $\boldsymbol{V} = \boldsymbol{V}_{\text{Eng}} \cup \boldsymbol{V}_{\text{Chi}}$ be the set of part-of-speech tags which belong to the verb category for either treebank. Then, we tie parameters for all part-of-speech tags in $\boldsymbol{V}$. We tested this joint model for each of TIEV, TIEN, TIEV&N, and TIEA. After running the inference algorithm which learns the two models jointly, we use unseen data to test each learned model separately.

Table 1 includes the results for these experiments. The performance on English improved significantly in the bilingual setting, achieving highest performance with TIEV&N. Performance with Chinese is also the highest in the bilingual setting, with TIEA and TIEV&N.

5 Future Work

In future work we plan to lexicalize the model, including a Bayesian grammar prior that accounts for the syntactic patterns of *words*. Nonparametric models (Teh, 2006) may be appropriate. We also believe that Bayesian discovery of cross-linguistic patterns is an exciting topic worthy of further exploration.

6 Conclusion

We described a Bayesian model that allows soft parameter tying among *any* weights in a probabilistic grammar. We used this model to improve unsupervised parsing accuracy on two different languages, English and Chinese, achieving state-of-the-art results. We also showed how our model can be effectively used to simultaneously learn grammars in two languages from non-parallel multilingual data.

Acknowledgments

This research was supported by NSF IIS-0836431. The authors thank the anonymous reviewers and Sylvia Rebholz for helpful comments.

[5] Haghighi et al. (2008) presented a technique to learn bilingual lexicons from two non-parallel monolingual corpora.

References

J. Aitchison. 1986. *The Statistical Analysis of Compositional Data*. Chapman and Hall, London.

D. M. Blei and J. D. Lafferty. 2006. Correlated topic models. In *Proc. of NIPS*.

D. M. Blei, A. Ng, and M. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

D. Burkett and D. Klein. 2008. Two languages are better than one (for syntactic parsing). In *Proc. of EMNLP*.

E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proc. of ACL*.

S. B. Cohen and N. A. Smith. 2009. Inference for probabilistic grammars with shared logistic normal distributions. Technical report, Carnegie Mellon University.

S. B. Cohen, K. Gimpel, and N. A. Smith. 2008. Logistic normal priors for unsupervised probabilistic grammar induction. In *NIPS*.

M. Collins. 2003. Head-driven statistical models for natural language processing. *Computational Linguistics*, 29:589–637.

I. Dagan. 1991. Two languages are more informative than one. In *Proc. of ACL*.

J. Eisner. 2002. Transformational priors over grammars. In *Proc. of EMNLP*.

J. R. Finkel, T. Grenager, and C. D. Manning. 2007. The infinite tree. In *Proc. of ACL*.

J. Goodman. 1996. Parsing algorithms and metrics. In *Proc. of ACL*.

A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proc. of ACL*.

W. P. Headden, M. Johnson, and D. McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proc. of NAACL-HLT*.

G. E. Hinton. 1999. Products of experts. In *Proc. of ICANN*.

M. Johnson, T. L. Griffiths, and S. Goldwater. 2006. Adaptor grammars: A framework for specifying compositional nonparameteric Bayesian models. In *NIPS*.

M. Johnson, T. L. Griffiths, and S. Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proc. of NAACL*.

M. Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *Proc. EMNLP-CoNLL*.

M. I. Jordan, Z. Ghahramani, T. S. Jaakola, and L. K. Saul. 1999. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

D. Klein and C. D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proc. of ACL*.

K. Kurihara and T. Sato. 2006. Variational Bayesian grammar induction for natural language. In *Proc. of ICGI*.

P. Liang, S. Petrov, M. Jordan, and D. Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In *Proc. of EMNLP*.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19:313–330.

D. A. Smith and N. A. Smith. 2004. Bilingual parsing with factored estimation: Using English to parse Korean. In *Proc. of EMNLP*, pages 49–56.

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

B. Snyder and R. Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proc. of ACL*.

Y. W. Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proc. of COLING-ACL*.

M. Wang, N. A. Smith, and T. Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for question answering. In *Proc. of EMNLP*.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comp. Ling.*, 23(3):377–404.

N. Xue, F. Xia, F.-D. Chiou, and M. Palmer. 2004. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 10(4):1–30.