The Shared Logistic Normal Distribution for Grammar Induction

Shay B. Cohen Noah A. Smith
Language Technologies Institute
Carnegie Mellon University
{scohen,nasmith}@cs.cmu.edu

Abstract

We present a shared logistic normal distribution as a Bayesian prior over probabilistic grammar weights. This approach generalizes the similar use of logistic normal distributions [3], enabling soft parameter tying during inference across different multinomials comprising the probabilistic grammar. We show that this model outperforms previous approaches on an unsupervised dependency grammar induction task.

1 Introduction

Probabilistic grammars are ubiquitous in natural language processing. This family of distributions, which includes hidden Markov models and probabilistic context free grammars, defines generative models using collections of multinomials that depend on each other through probabilistic conditioning.

There has been an increased interest in the use of Bayesian methods as applied to probabilistic grammars for NLP, including part-of-speech tagging [6, 15], phrase-structure parsing [4, 9, 12], and combinations of models [5]. In Bayesian-minded work with probabilistic grammars, a common thread is the use of a Dirichlet prior for the underlying multinomials, because as the conjugate prior for the multinomial, it bestows computational feasibility, and can also encourage sparsity in the learned grammar [9].

Yet overcoming the limitations of conjugacy can be crucial for performance in a Bayesian setting. In [2], Blei and Lafferty replaced the Dirichlet distribution with a logistic normal (LN) distribution as a choice for topic distributions; in [3] we applied a collection of LN distributions to probabilistic grammars in an empirical Bayesian setting. This model enables parameter tying among grammar weights that belong to the same multinomial distribution, an ability that has proven to be useful in a non-Bayesian setting as well [4]. With a generative dependency parsing model, the relationships one might expect to find when doing inference using this model are ones such as "the probability of pronouns and nouns being attached to a verb on the left are correlated positively." This model led to a significant increase in performance over state-of-the-art performance for unsupervised dependency parsing.

The model in [3] does not allow us to tie parameters between *different* multinomials of the same model. This means, for example, that we cannot tie the probabilities of nouns being attached to a verb or a modal verb, where each parent implies a different probabilistic conditioning. In this paper, we extend the model to allow such relationships. Our model is based on a definition of a new type of distribution over collections of multinomials, the *shared* logistic normal (SLN) distribution, which is a generalization of the LN distribution.

2 Probabilistic Grammars

In general, a probabilistic grammar defines the joint probability of a string $\mathbf{x}$ and a grammatical derivation $\mathbf{y}$ using a step-by-step process for building a derivation from that grammar:

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

$f_{k,i}$ is a function that "counts" the number of times the kth distribution's ith event occurs in the derivation.

The parameters $\boldsymbol{\theta}$ are a collection of K multinomials $\langle \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K \rangle$, the kth of which includes N_k events. HMMs and vanilla PCFGs are the best known probabilistic grammars, but there are others. In this paper, we focus in the "dependency model with valence," a probabilistic grammar for dependency parsing first proposed in [11] and extended in [14].

Let $\mathbf{x} = \langle x_1, x_2, ..., x_n \rangle$ be a sentence (here, as in prior work, represented as a sequence of part-of-speech tags). 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:

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

where $\mathrm{first}_{\mathbf{y}}(j)$ 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 entire tree is given by $p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta}) = P(\mathbf{y}^{(0)} \mid \$, \boldsymbol{\theta})$. The parameters $\boldsymbol{\theta}$ are the multinomial distributions $\theta_{\mathrm{s}}(\cdot \mid \cdot, \cdot, \cdot)$ and $\theta_{\mathrm{c}}(\cdot \mid \cdot, \cdot)$. To follow the general setting of Eq. 1, we index these distributions as $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K$.

3 The Shared Logistic Normal Prior

Our aim is to define a prior distribution over $\boldsymbol{\theta}$, the collection multinomial distributions that parameterize the probabilistic grammar. In [3] we showed how a logistic normal prior was able to capture linguistically motivated correlations between different $\theta_{k,i}$ and greatly improve performance. In particular, for a given k, the logistic normal grammar models covariance between different $\theta_{k,i}$, for different i. In our model, this captured the effect that a category likely to take arguments from one category will tend also to accept arguments from related categories.

Here we go farther, permitting parameter tying across any of the $\theta_{k,i}$. Each $\theta_{k,i}$ will be defined by

$$\theta_{k,i} = \exp\left(\frac{1}{|\mathfrak{I}_{k,i}|} \sum_{j \in \mathfrak{I}_{k,i}} \eta_j\right) \Big/ \sum_{i'=1}^{N_k} \exp\left(\frac{1}{|\mathfrak{I}_{k,i'}|} \sum_{j \in \mathfrak{I}_{k,i'}} \eta_j\right) \quad (3)$$

where $\boldsymbol{\eta}$ is a (random) vector in $\mathbb{R}^D$ and $\mathfrak{I}_{k,i} \subseteq \{1, ..., D\}$. Each η_j is uniquely associated with a single $\theta_{k,i}$, and each $\theta_{k,i}$ is defined by taking a(n exponentiated) geometric average of its corresponding η_js. Such a distribution generalizes the *partitioned logistic normal* (PLN) from Aitchison [1] and overcomes an otherwise problematic complexity issue with PLN when doing inference, as we explain later.

To model covariance among arbitrary $\theta_{k,i}$, we allow pairwise covariance between different η_j; $\boldsymbol{\eta}$ is a random variable chosen according to a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. It is convenient to think of each η_j as a weight associated with a unique event's probability. By letting different η_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. Eq. 3 also implies that we multiply together several multinomials in a kind of product of experts [7].

As mentioned above, Eq. 3 is a generalization of the *partitioned logistic normal* (PLN) distribution by [1]. With PLNs, we have that $|\mathcal{I}_{k,i}| = 1$. The logistic normal grammar of [3] is also special case where we add an additional constraint to $|\mathcal{I}_{k,i}| = 1$: the covariance matrix $\boldsymbol{\Sigma}$ is fixed at 0 for any pair η_j, $\eta_{j'}$ that correspond to $\theta_{k,i}$, $\theta_{k',i'}$, respectively, whenever $k \neq k'$. This can be accomplished by choosing different portions of $\boldsymbol{\eta}$ from different, separate normal distributions, whose parameters are learned.

In this paper, we will again break $\boldsymbol{\eta}$ into several subvectors, assuming different, independent normal distributions over each. This is important for tractability and to keep the number of parameters in the model from growing beyond what can reasonably be learned; it is equivalent to fixing many covariances at 0. Deciding which η_j may covary, or, equivalently, which $\theta_{k,i}$ are connected to each other through inclusion of covarying η_js, is a crucial decision in designing a model. After defining the covariance relationships, we end up with a distribution over a collection of multinomials, encoding the parameters of the generative probabilistic grammar. We denote a sample from this distribution by $\boldsymbol{\theta} \sim \text{SLN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathcal{I})$, where SLN stands for *shared logistic normal*, $(\boldsymbol{\mu}, \Sigma)$ encode the *normal experts* and $\mathcal{I}$ encodes the *partition structure* of $\boldsymbol{\theta}$.

With the shared logistic normal distribution defined, our model follows naturally. The generative story for this model is as following:

1. Generate $\boldsymbol{\theta} \sim \text{SLN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathcal{I})$, 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).

Our inference algorithm is a modified version of the variational inference algorithm in [3] which is based on the one in [2]. We will assume that $\mathcal{I}$ is fixed together with cells in Σ that will be kept 0, and then use variational EM for estimating the hyperparameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. After learning $\boldsymbol{\mu}$, apply a softmax function to map it to the probability simplexes, based on the partition structure $\mathcal{I}$.

4 Experiments

Data Following the setting in [10], we experimented using part-of-speech sequences from the *Wall Street Journal* Penn Treebank [13], stripped of words and punctuation. We follow standard parsing conventions and train on sections 2–21 (sentences of length shorter than 10), tune on section 22, and report final results on section 23.

Evaluation For the SLN experiments, we run the inference algorithm (variational EM) for a fixed number of 10 iterations.[1] After extracting a point estimate $\boldsymbol{\theta}$, we predict $\mathbf{y}$ for unseen test data (by parsing with the probabilistic grammar) and report the fraction of words whose predicted parent matches the gold standard corpus, known as attachment accuracy. Parses were selected using minimum Bayes risk parsing to minimize expected attachment error.

Settings Our experiments compare four settings for estimating the probabilistic grammar. For full information about initialization and stopping criterion, see [3]. **Dirichlet** uses variational Bayes EM with a Dirichlet prior, in an empirical Bayes setting (i.e. estimate the Dirichlet parameters). **LN** uses variational Bayes EM with a logistic normal prior (as in [3]); **SLN** (TieVerbs) uses variational Bayes EM with a SLN prior that softly ties together all multinomials that condition on a verb parent, per direction (left or right) and first attribute; **SLN** (TieNouns) is analogous, but for *noun* parents; and **SLN** (TieVerbsAndNouns) uses variational Bayes EM with a SLN prior and the union of the previous two SLN models' partition structures.

Results Table 1 gives the results. The best performance is attained using SLN with nouns and verbs grouped through $\mathcal{I}$.

[1] For the SLN experiments, we do not check against unseen data to monitor convergence of likelihood, but instead run variational EM for a fixed number of iterations. With EM, accuracy decreases in the first few iterations [8].

	attachment accuracy (%)		
	≤ 10	≤ 20	all
Attach-Right	38.4	33.4	31.7
EM [11]	46.1	39.9	35.9
Dirichlet (variational inference with emp. Bayes)	46.1	40.6	36.9
LN (Cohen et al. 2008)	59.4	45.9	40.5
SLN (TieVerbs)	60.2	46.2	40.0
SLN (TieNouns)	60.2	46.7	40.9
SLN (TieVerbsAndNouns)	61.3	47.4	41.4

Table 1: Attachment accuracy of different models, on test data from the Penn 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 \$. Covariance matrices were initialized using a coarser tag set like in [3].

5 Conclusion and Future Directions

We presented a generative model for probabilistic grammars based on a newly defined distribution, the shared logistic normal, allowing flexible parameter tying in the model. We used the model to attain state-of-the-art unsupervised dependency parsing results for English. Future directions include learning the partition structure and employing the model for other kinds of probabilistic grammars and other kinds of data.

Acknowledgments This work was made possible by an IBM faculty award, NSF grants IIS-0713265 and IIS-0836431 to the second author and computational resources provided by Yahoo.

References

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

[2] D. M. Blei and J. D. Lafferty. Correlated topic models. In *NIPS*, 2006.

[3] S. B. Cohen, K. Gimpel, and N. A. Smith. Unsupervised Bayesian parameter estimation for probabilistic grammars. In *NIPS*, 2008.

[4] J. Eisner. Transformational priors over grammars. In *Proc. of EMNLP*, 2002.

[5] J. R. Finkel, C. D. Manning, and A. Y. Ng. Solving the problem of cascading errors: Approximate Bayesian inference for linguistic annotation pipelines. In *Proc. of EMNLP*, 2006.

[6] S. Goldwater and T. L. Griffiths. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proc. of ACL*, 2007.

[7] G. E. Hinton. Products of experts. In *Proc. of ICANN*, 1999.

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

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

[10] D. Klein and C. D. Manning. A generative constituent-context model for improved grammar induction. In *Proc. of ACL*, 2002.

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

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

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

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

[15] K. Toutanova and M. Johnson. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *NIPS*, 2007.