Journal of Machine Learning Research 15 (2014) 2399-2449 Submitted 1/14; Published 7/14

Spectral Learning of Latent-Variable PCFGs: Algorithms and Sample Complexity

Shay B. Cohen SCOHEN@INF.ED.AC.UK
School of Informatics
University of Edinburgh
Edinburgh, EH8 9LE, UK

Karl Stratos STRATOS@CS.COLUMBIA.EDU
Michael Collins MCOLLINS@CS.COLUMBIA.EDU
Department of Computer Science
Columbia University
New York, NY 10027, USA

Dean P. Foster DEAN@FOSTER.NET
Yahoo! Labs
New York, NY 10018, USA

Lyle Ungar UNGAR@CIS.UPENN.EDU
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA

Editor: Alexander Clark

Abstract

We introduce a spectral learning algorithm for latent-variable PCFGs (Matsuzaki et al., 2005; Petrov et al., 2006). Under a separability (singular value) condition, we prove that the method provides statistically consistent parameter estimates. Our result rests on three theorems: the first gives a tensor form of the inside-outside algorithm for PCFGs; the second shows that the required tensors can be estimated directly from training examples where hidden-variable values are missing; the third gives a PAC-style convergence bound for the estimation method.

Keywords: latent-variable PCFGs, spectral learning algorithms

1. Introduction

Statistical models with hidden or latent variables are of great importance in natural language processing, speech, and many other fields. The EM algorithm is a remarkably successful method for parameter estimation within these models: it is simple, it is often relatively efficient, and it has well understood formal properties. It does, however, have a major limitation: it has no guarantee of finding the global optimum of the likelihood function. From a theoretical perspective, this means that the EM algorithm is not guaranteed to give statistically consistent parameter estimates. From a practical perspective, problems with local optima can be difficult to deal with.

Recent work has introduced a polynomial-time learning algorithm for an important case of hidden-variable models: hidden Markov models (Hsu et al., 2009). This algorithm uses a spectral method: that is, an algorithm based on eigenvector decompositions of linear systems, in particular singular value decomposition (SVD). In the general case, learning of HMMs is intractable (e.g., see Terwijn, 2002). The spectral method finesses the problem of intractability by assuming separability conditions. More precisely, the algorithm of Hsu et al. (2009) has a sample complexity that is polynomial in $1/\sigma$, where σ is the minimum singular value of an underlying decomposition. The HMM learning algorithm is not susceptible to problems with local maxima.

In this paper we derive a spectral algorithm for learning of latent-variable PCFGs (L-PCFGs) (Petrov et al., 2006; Matsuzaki et al., 2005). L-PCFGs have been shown to be a very effective model for natural language parsing. Under a condition on singular values in the underlying model, our algorithm provides consistent parameter estimates; this is in contrast with previous work, which has used the EM algorithm for parameter estimation, with the usual problems of local optima.

The parameter estimation algorithm (see Figure 7) is simple and efficient. The first step is to take an SVD of the training examples, followed by a projection of the training examples down to a low-dimensional space. In a second step, empirical averages are calculated on the training examples, followed by standard matrix operations. On test examples, tensor-based variants of the inside-outside algorithm (Figures 4 and 5) can be used to calculate probabilities and marginals of interest.

Our method depends on the following results:

- *Tensor form of the inside-outside algorithm.* Section 6.1 shows that the inside-outside algorithm for L-PCFGs can be written using tensors and tensor products. Theorem 3 gives conditions under which the tensor form calculates inside and outside terms correctly.

- *Observable representations.* Section 7.2 shows that under a singular-value condition, there is an *observable form* for the tensors required by the inside-outside algorithm. By an observable form, we follow the terminology of Hsu et al. (2009) in referring to quantities that can be estimated directly from data where values for latent variables are unobserved. Theorem 6 shows that tensors derived from the observable form satisfy the conditions of Theorem 3.

- *Estimating the model.* Section 8 gives an algorithm for estimating parameters of the observable representation from training data. Theorem 8 gives a sample complexity result, showing that the estimates converge to the true distribution at a rate of $1/\sqrt{M}$ where M is the number of training examples.

The algorithm is strikingly different from the EM algorithm for L-PCFGs, both in its basic form, and in its consistency guarantees. The techniques developed in this paper are quite general, and should be relevant to the development of spectral methods for estimation in other models in NLP, for example alignment models for translation, synchronous PCFGs, and so on. The tensor form of the inside-outside algorithm gives a new view of basic calculations in PCFGs, and may itself lead to new models.

In this paper we derive the basic algorithm, and the theory underlying the algorithm. In a companion paper (Cohen et al., 2013), we describe experiments using the algorithm to learn an L-PCFG for natural language parsing. In these experiments the spectral algorithm gives models that are as accurate as the EM algorithm for learning in L-PCFGs. It is significantly more efficient than the EM algorithm on this problem (9h52m of training time vs. 187h12m), because after an SVD operation it requires a single pass over the data, whereas EM requires around 20-30 passes before converging to a good solution.

2. Related Work

The most common approach for statistical learning of models with latent variables is the expectation-maximization (EM) algorithm (Dempster et al., 1977). Under mild conditions, the EM algorithm is guaranteed to converge to a local maximum of the log-likelihood function. This is, however, a relatively weak guarantee; there are in general no guarantees of consistency for the EM algorithm, and no guarantees of sample complexity, for example within the PAC framework (Valiant, 1984). This has led a number of researchers to consider alternatives to the EM algorithm, which do have PAC-style guarantees.

One focus of this work has been on the problem of learning Gaussian mixture models. In early work, Dasgupta (1999) showed that under separation conditions for the underlying Gaussians, an algorithm with PAC guarantees can be derived. For more recent work in this area, see for example Vempala and Wang (2004), and Moitra and Valiant (2010). These algorithms avoid the issues of local maxima posed by the EM algorithm.

Another focus has been on spectral learning algorithms for hidden Markov models (HMMs) and related models. This work forms the basis for the L-PCFG learning algorithms described in this paper. This line of work started with the work of Hsu et al. (2009), who developed a spectral learning algorithm for HMMs which recovers an HMM's parameters, up to a linear transformation, using singular value decomposition and other simple matrix operations. The algorithm builds on the idea of observable operator models for HMMs due to Jaeger (2000). Following the work of Hsu et al. (2009), spectral learning algorithms have been derived for a number of other models, including finite state transducers (Balle et al., 2011); split-head automaton grammars (Luque et al., 2012); reduced rank HMMs in linear dynamical systems (Siddiqi et al., 2010); kernel-based methods for HMMs (Song et al., 2010); and tree graphical models (Parikh et al., 2011; Song et al., 2011). There are also spectral learning algorithms for learning PCFGs in the unsupervised setting (Bailly et al., 2013).

Foster et al. (2012) describe an alternative algorithm to that of Hsu et al. (2009) for learning of HMMs, which makes use of tensors. Our work also makes use of tensors, and is closely related to the work of Foster et al. (2012); it is also related to the tensor-based approaches for learning of tree graphical models described by Parikh et al. (2011) and Song et al. (2011). In related work, Dhillon et al. (2012) describe a tensor-based method for dependency parsing.

Bailly et al. (2010) describe a learning algorithm for weighted (probabilistic) tree automata that is closely related to our own work. Our approach leverages functions ϕ and ψ that map inside and outside trees respectively to feature vectors (see Section 7.2): for example, $\phi(t)$ might track the context-free rule at the root of the inside tree t, or features

corresponding to larger tree fragments. Cohen et al. (2013) give definitions of ϕ and ψ used in parsing experiments with L-PCFGs. In the special case where ϕ and ψ are identity functions, specifying the entire inside or outside tree, the learning algorithm of Bailly et al. (2010) is the same as our algorithm. However, our work differs from that of Bailly et al. (2010) in several important respects. The generalization to allow arbitrary functions ϕ and ψ is important for the success of the learning algorithm, in both a practical and theoretical sense. The inside-outside algorithm, derived in Figure 5, is not presented by Bailly et al. (2010), and is critical in deriving marginals used in parsing. Perhaps most importantly, the analysis of sample complexity, given in Theorem 8 of this paper, is much tighter than the sample complexity bound given by Bailly et al. (2010). The sample complexity bound in theorem 4 of Bailly et al. (2010) suggests that the number of samples required to obtain $|\hat{p}(t) - p(t)| \leqslant \epsilon$ for some tree t of size N, and for some value ϵ, is *exponential* in N. In contrast, we show that the number of samples required to obtain $\sum_t |\hat{p}(t) - p(t)| \leqslant \epsilon$ where the sum is over *all* trees of size N is polynomial in N. Thus our bound is an improvement in a couple of ways: first, it applies to a sum over all trees of size N, a set of exponential size; second, it is polynomial in N.

Spectral algorithms are inspired by the method of moments, and there are latent-variable learning algorithms that use the method of moments, without necessarily resorting to spectral decompositions. Most relevant to this paper is the work in Cohen and Collins (2014) for estimating L-PCFGs, inspired by the work by Arora et al. (2013).

3. Notation

Given a matrix A or a vector v, we write $A^\top$ or $v^\top$ for the associated transpose. For any integer $n \geqslant 1$, we use $[n]$ to denote the set $\{1, 2, \ldots n\}$.

We use $\mathbb{R}^{m \times 1}$ to denote the space of m-dimensional column vectors, and $\mathbb{R}^{1 \times m}$ to denote the space of m-dimensional row vectors. We use $\mathbb{R}^m$ to denote the space of m-dimensional vectors, where the vector in question can be either a row or column vector. For any row or column vector $y \in \mathbb{R}^m$, we use $\text{diag}(y)$ to refer to the $(m \times m)$ matrix with diagonal elements equal to y_h for $h = 1 \ldots m$, and off-diagonal elements equal to 0. For any statement Γ, we use $[\![\Gamma]\!]$ to refer to the indicator function that is 1 if Γ is true, and 0 if Γ is false. For a random variable X, we use $\mathbf{E}[X]$ to denote its expected value.

We will make use of tensors of rank 3:

Definition 1 *A tensor $C \in \mathbb{R}^{(m \times m \times m)}$ is a set of m^3 parameters $C_{i,j,k}$ for $i, j, k \in [m]$. Given a tensor C, and vectors $y^1 \in \mathbb{R}^m$ and $y^2 \in \mathbb{R}^m$, we define $C(y^1, y^2)$ to be the m-dimensional row vector with components*

$$[C(y^1, y^2)]_i = \sum_{j \in [m], k \in [m]} C_{i,j,k} y^1_j y^2_k.$$

Hence C can be interpreted as a function $C : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^{1 \times m}$ that maps vectors y^1 and y^2 to a row vector $C(y^1, y^2) \in \mathbb{R}^{1 \times m}$.

In addition, we define the tensor $C_{(1,2)} \in \mathbb{R}^{(m \times m \times m)}$ for any tensor $C \in \mathbb{R}^{(m \times m \times m)}$ to be the function $C_{(1,2)} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^{m \times 1}$ defined as

$$[C_{(1,2)}(y^1, y^2)]_k = \sum_{i \in [m], j \in [m]} C_{i,j,k} y^1_i y^2_j.$$

Similarly, for any tensor C we define $C_{(1,3)} : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^{m \times 1}$ as

$$[C_{(1,3)}(y^1, y^2)]_j = \sum_{i \in [m], k \in [m]} C_{i,j,k} y^1_i y^2_k.$$

Note that $C_{(1,2)}(y^1, y^2)$ and $C_{(1,3)}(y^1, y^2)$ are both **column** *vectors.*

For vectors $x, y, z \in \mathbb{R}^m$, $xy^\top z^\top$ is the tensor $D \in \mathbb{R}^{m \times m \times m}$ where $D_{i,j,k} = x_i y_j z_k$ (this is analogous to the outer product: $[xy^\top]_{i,j} = x_i y_j$).

We use $||\ldots||_F$ to refer to the Frobenius norm for matrices or tensors: for a matrix A, $||A||_F = \sqrt{\sum_{i,j}(A_{i,j})^2}$, for a tensor C, $||C||_F = \sqrt{\sum_{i,j,k}(C_{i,j,k})^2}$. For a matrix A we use $||A||_{2,o}$ to refer to the operator (spectral) norm, $||A||_{2,o} = \max_{x \neq 0} ||Ax||_2 / ||x||_2$.

4. L-PCFGs

In this section we describe latent-variable PCFGs (L-PCFGs), as used for example by Matsuzaki et al. (2005) and Petrov et al. (2006). We first give the basic definitions for L-PCFGs, and then describe the underlying motivation for them.

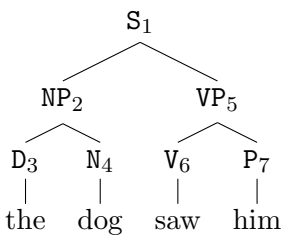
4.1 Basic Definitions

An L-PCFG is an 8-tuple $(\mathcal{N}, \mathcal{I}, \mathcal{P}, m, n, t, q, \pi)$ where:

- $\mathcal{N}$ is the set of non-terminal symbols in the grammar. $\mathcal{I} \subset \mathcal{N}$ is a finite set of *in-terminals*. $\mathcal{P} \subset \mathcal{N}$ is a finite set of *pre-terminals*. We assume that $\mathcal{N} = \mathcal{I} \cup \mathcal{P}$, and $\mathcal{I} \cap \mathcal{P} = \varnothing$. Hence we have partitioned the set of non-terminals into two subsets.
- $[m]$ is the set of possible hidden states.
- $[n]$ is the set of possible words.
- For all $a \in \mathcal{I}$, $b \in \mathcal{N}$, $c \in \mathcal{N}$, $h_1, h_2, h_3 \in [m]$, we have a context-free rule $a(h_1) \rightarrow b(h_2)\ c(h_3)$.
- For all $a \in \mathcal{P}$, $h \in [m]$, $x \in [n]$, we have a context-free rule $a(h) \rightarrow x$.
- For all $a \in \mathcal{I}, b, c \in \mathcal{N}$, and $h_1, h_2, h_3 \in [m]$, we have a parameter $t(a \rightarrow b\ c, h_2, h_3 | h_1, a)$.
- For all $a \in \mathcal{P}$, $x \in [n]$, and $h \in [m]$, we have a parameter $q(a \rightarrow x | h, a)$.
- For all $a \in \mathcal{I}$ and $h \in [m]$, we have a parameter $\pi(a, h)$ which is the probability of non-terminal a paired with hidden variable h being at the root of the tree.

Note that each in-terminal $a \in \mathcal{I}$ is always the left-hand-side of a binary rule $a \rightarrow b\ c$; and each pre-terminal $a \in \mathcal{P}$ is always the left-hand-side of a rule $a \rightarrow x$. Assuming that the non-terminals in the grammar can be partitioned this way is relatively benign, and makes the estimation problem cleaner.

For convenience we define the set of possible "skeletal rules" as $\mathcal{R} = \{a \rightarrow b\ c : a \in \mathcal{I}, b \in \mathcal{N}, c \in \mathcal{N}\}$.



$r_1 = \texttt{S} \rightarrow \texttt{NP VP}$
$r_2 = \texttt{NP} \rightarrow \texttt{D N}$
$r_3 = \texttt{D} \rightarrow \texttt{the}$
$r_4 = \texttt{N} \rightarrow \texttt{dog}$
$r_5 = \texttt{VP} \rightarrow \texttt{V P}$
$r_6 = \texttt{V} \rightarrow \texttt{saw}$
$r_7 = \texttt{P} \rightarrow \texttt{him}$

Figure 1: s-tree, and its sequence of rules. (For convenience we have numbered the nodes in the tree.)

These definitions give a PCFG, with rule probabilities

$$p(a(h_1) \rightarrow b(h_2)\ c(h_3)|a(h_1)) = t(a \rightarrow b\ c, h_2, h_3|h_1, a),$$

and

$$p(a(h) \rightarrow x|a(h)) = q(a \rightarrow x|h, a).$$

Remark 2 *In the previous paper on this work (Cohen et al., 2012), we considered an L-PCFG model where*

$$p(a(h_1) \rightarrow b(h_2)\ c(h_3)|a(h_1)) = p(a \rightarrow b\ c|h_1, a) \times p(h_2|h_1, a \rightarrow b\ c) \times p(h_3|h_1, a \rightarrow b\ c)$$

In this model the random variables h_2 and h_3 are assumed to be conditionally independent given h_1 and $a \rightarrow b\ c$.

In this paper we consider a model where

$$p(a(h_1) \rightarrow b(h_2)\ c(h_3)|a(h_1)) = t(a \rightarrow b\ c, h_2, h_3, |h_1, a). \tag{1}$$

That is, we do **not** *assume that the random variables h_2 and h_3 are independent when conditioning on h_1 and $a \rightarrow b\ c$. This is also the model considered by Matsuzaki et al. (2005) and Petrov et al. (2006).*

Note however that the algorithms in this paper are the same as those in Cohen et al. (2012): we have simply proved that the algorithms give consistent estimators for the model form in Eq. 1.

As in usual PCFGs, the probability of an entire tree is calculated as the product of its rule probabilities. We now give more detail for these calculations.

An L-PCFG defines a distribution over parse trees as follows. A *skeletal tree* (s-tree) is a sequence of rules $r_1 \ldots r_N$ where each r_i is either of the form $a \rightarrow b\ c$ or $a \rightarrow x$. The rule sequence forms a top-down, left-most derivation under a CFG with skeletal rules. See Figure 1 for an example.

A *full tree* consists of an s-tree $r_1 \ldots r_N$, together with values $h_1 \ldots h_N$. Each h_i is the value for the hidden variable for the left-hand-side of rule r_i. Each h_i can take any value in $[m]$.

Define a_i to be the non-terminal on the left-hand-side of rule r_i. For any $i \in [N]$ such that $a_i \in \mathcal{I}$ (i.e., a_i is an in-terminal, and rule r_i is of the form $a \rightarrow b\ c$) define $h_i^{(2)}$ to be the hidden variable value associated with the left child of the rule r_i, and $h_i^{(3)}$ to be the hidden variable value associated with the right child. The probability mass function (PMF) over full trees is then

$$p(r_1 \ldots r_N, h_1 \ldots h_N) = \pi(a_1, h_1) \times \prod_{i:a_i \in \mathcal{I}} t(r_i, h_i^{(2)}, h_i^{(3)} | h_i, a_i) \times \prod_{i:a_i \in \mathcal{P}} q(r_i | h_i, a_i). \quad (2)$$

The PMF over s-trees is $p(r_1 \ldots r_N) = \sum_{h_1 \ldots h_N} p(r_1 \ldots r_N, h_1 \ldots h_N)$.

In the remainder of this paper, we make use of a matrix form of parameters of an L-PCFG, as follows:

- For each $a \rightarrow b\ c \in \mathcal{R}$, we define $T^{a \rightarrow b\ c} \in \mathbb{R}^{m \times m \times m}$ to be the tensor with values

$$T^{a \rightarrow b\ c}_{h_1, h_2, h_3} = t(a \rightarrow b\ c, h_2, h_3 | a, h_1).$$

- For each $a \in \mathcal{P}$, $x \in [n]$, we define $q_{a \rightarrow x} \in \mathbb{R}^{1 \times m}$ to be the row vector with values

$$[q_{a \rightarrow x}]_h = q(a \rightarrow x | h, a)$$

  for $h = 1, 2, \ldots m$.

- For each $a \in \mathcal{I}$, we define the column vector $\pi^a \in \mathbb{R}^{m \times 1}$ where $[\pi^a]_h = \pi(a, h)$.

4.2 Application of L-PCFGs to Natural Language Parsing

L-PCFGs have been shown to be a very useful model for natural language parsing (Matsuzaki et al., 2005; Petrov et al., 2006). In this section we describe the basic approach.

We assume a training set consisting of sentences paired with parse trees, which are similar to the skeletal tree shown in Figure 1. A naive approach to parsing would simply read off a PCFG from the training set: the resulting grammar would have rules such as

```
S → NP VP
NP → D N
VP → V NP
D → the
N → dog
```

and so on. Given a test sentence, the most likely parse under the PCFG can be found using dynamic programming algorithms.

Unfortunately, simple "vanilla" PCFGs induced from treebanks such as the Penn treebank (Marcus et al., 1993) typically give very poor parsing performance. A critical issue is that the set of non-terminals in the resulting grammar (`S, NP, VP, PP, D, N`, etc.) is often quite small. The resulting PCFG therefore makes very strong independence assumptions, failing to capture important statistical properties of parse trees.

In response to this issue, a number of PCFG-based models have been developed which make use of grammars with *refined* non-terminals. For example, in lexicalized models

(Collins, 1997; Charniak, 1997), non-terminals such as S are replaced with non-terminals such as S-sleeps: the non-terminals track some lexical item (in this case *sleeps*), in addition to the syntactic category. For example, the parse tree in Figure 1 would include rules

$$\begin{aligned}
\texttt{S-saw} &\rightarrow \texttt{NP-dog VP-saw} \\
\texttt{NP-dog} &\rightarrow \texttt{D-the N-dog} \\
\texttt{VP-saw} &\rightarrow \texttt{V-saw P-him} \\
\texttt{D-the} &\rightarrow \texttt{the} \\
\texttt{N-dog} &\rightarrow \texttt{dog} \\
\texttt{V-saw} &\rightarrow \texttt{saw} \\
\texttt{P-him} &\rightarrow \texttt{him}
\end{aligned}$$

In this case the number of non-terminals in the grammar increases dramatically, but with appropriate smoothing of parameter estimates lexicalized models perform at much higher accuracy than vanilla PCFGs.

As another example, Johnson describes an approach where non-terminals are refined to also include the non-terminal one level up in the tree; for example rules such as

$$\texttt{S} \rightarrow \texttt{NP VP}$$

are replaced by rules such as

$$\texttt{S-ROOT} \rightarrow \texttt{NP-S VP-S}$$

Here NP-S corresponds to an NP non-terminal whose parent is S; VP-S corresponds to a VP whose parent is S; S-ROOT corresponds to an S which is at the root of the tree. This simple modification leads to significant improvements over a vanilla PCFG.

Klein and Manning (2003) develop this approach further, introducing annotations corresponding to parents and siblings in the tree, together with other information, resulting in a parser whose performance is just below the lexicalized models of Collins (1997) and Charniak (1997).

The approaches of Collins (1997), Charniak (1997), Johnson, and Klein and Manning (2003) all use hand-constructed rules to enrich the set of non-terminals in the PCFG. A natural question is whether refinements to non-terminals can be learned automatically. Matsuzaki et al. (2005) and Petrov et al. (2006) addressed this question through the use of L-PCFGs in conjunction with the EM algorithm. The basic idea is to allow each non-terminal in the grammar to have m possible *latent* values. For example, with $m = 8$ we would replace the non-terminal S with non-terminals S-1, S-2, ..., S-8, and we would replace rules such as

$$\texttt{S} \rightarrow \texttt{NP VP}$$

with rules such as

$$\texttt{S-4} \rightarrow \texttt{NP-3 VP-2}$$

The latent values are of course unobserved in the training data (the treebank), but they can be treated as latent variables in a PCFG-based model, and the parameters of the model can

be estimated using the EM algorithm. More specifically, given training examples consisting of skeletal trees of the form $t^{(i)} = (r_1^{(i)}, r_2^{(i)}, \ldots, r_{N_i}^{(i)})$, for $i = 1 \ldots M$, where N_i is the number of rules in the i'th tree, the log-likelihood of the training data is

$$\sum_{i=1}^{M} \log p(r_1^{(i)} \ldots r_{N_i}^{(i)}) = \sum_{i=1}^{M} \log \sum_{h_1 \ldots h_{N_i}} p(r_1^{(i)} \ldots r_{N_i}^{(i)}, h_1 \ldots h_{N_i})$$

where $p(r_1^{(i)} \ldots r_{N_i}^{(i)}, h_1 \ldots h_{N_i})$ is as defined in Eq. 2. The EM algorithm is guaranteed to converge to a local maximum of the log-likelihood function. Once the parameters of the L-PCFG have been estimated, the algorithm of Goodman (1996) can be used to parse test-data sentences using the L-PCFG: see Section 4.3 for more details. Matsuzaki et al. (2005) and Petrov et al. (2006) show very good performance for these methods.

4.3 Basic Algorithms for L-PCFGs: Variants of the Inside-Outside Algorithm

Variants of the inside-outside algorithm (Baker, 1979) can be used for basic calculations in L-PCFGs, in particular for calculations that involve marginalization over the values for the hidden variables.

To be more specific, given an L-PCFG, two calculations are central:

1. For a given s-tree $r_1 \ldots r_N$, calculate $p(r_1 \ldots r_N) = \sum_{h_1 \ldots h_N} p(r_1 \ldots r_N, h_1 \ldots h_N)$.
2. For a given input sentence $x = x_1 \ldots x_N$, calculate the marginal probabilities

$$\mu(a, i, j) = \sum_{\tau \in \mathcal{T}(x) : (a,i,j) \in \tau} p(\tau)$$

for each non-terminal $a \in \mathcal{N}$, for each (i, j) such that $1 \leqslant i \leqslant j \leqslant N$. Here $\mathcal{T}(x)$ denotes the set of all possible s-trees for the sentence x, and we write $(a, i, j) \in \tau$ if non-terminal a spans words $x_i \ldots x_j$ in the parse tree τ.

The marginal probabilities have a number of uses. Perhaps most importantly, for a given sentence $x = x_1 \ldots x_N$, the parsing algorithm of Goodman (1996) can be used to find

$$\arg \max_{\tau \in \mathcal{T}(x)} \sum_{(a,i,j) \in \tau} \mu(a, i, j).$$

This is the parsing algorithm used by Petrov et al. (2006), for example.[1] In addition, we can calculate the probability for an input sentence, $p(x) = \sum_{\tau \in \mathcal{T}(x)} p(\tau)$, as $p(x) = \sum_{a \in \mathcal{I}} \mu(a, 1, N)$.

Figures 2 and 3 give the conventional (as opposed to tensor) form of inside-outside algorithms for these two problems. In the next section we describe the tensor form. The algorithm in Figure 2 uses dynamic programming to compute

$$p(r_1 \ldots r_N) = \sum_{h_1 \ldots h_N} p(r_1 \ldots r_N, h_1 \ldots h_N)$$

1. Note that finding $\arg\max_{\tau \in \mathcal{T}(x)} p(\tau)$, where $p(\tau) = \sum_{h_1 \ldots h_N} p(\tau, h_1 \ldots h_N)$, is NP hard, hence the use of Goodman's algorithm. Goodman's algorithm minimizes a different loss function when parsing: it minimizes the expected number of spans which are incorrect in the parse tree according to the underlying L-PCFG. We use it while restricting the output tree to be valid under the PCFG grammar extracted from the treebank. There are variants of Goodman's algorithm that do not follow this restriction.

Inputs: s-tree $r_1 \ldots r_N$, L-PCFG $(\mathcal{N}, \mathcal{I}, \mathcal{P}, m, n, t, q, \pi)$, with parameters

- $t(a \rightarrow b\ c, h_2, h_3 | h_1, a)$ for all $a \rightarrow b\ c \in \mathcal{R}$, $h_1, h_2, h_3 \in [m]$.
- $q(a \rightarrow x | h, a)$ for all $a \in \mathcal{P}, x \in [n], h \in [m]$
- $\pi(a, h)$ for all $a \in \mathcal{I}$, $h \in [m]$.

Algorithm: (calculate the b^i terms bottom-up in the tree)

- For all $i \in [N]$ such that $a_i \in \mathcal{P}$, for all $h \in [m]$, $b^i_h = q(r_i | h, a_i)$
- For all $i \in [N]$ such that $a_i \in \mathcal{I}$, for all $h \in [m]$, $b^i_h = \sum_{h_2, h_3} t(r_i, h_2, h_3 | h, a_i) b^\beta_{h_2} b^\gamma_{h_3}$ where β is the index of the left child of node i in the tree, and γ is the index of the right child.

Return: $\sum_h b^1_h \pi(a_1, h) = p(r_1 \ldots r_N)$

Figure 2: The conventional inside-outside algorithm for calculation of $p(r_1 \ldots r_N)$.

for a given parse tree $r_1 \ldots r_N$. The algorithm in Figure 3 uses dynamic programming to compute marginal terms.

5. Roadmap

The next three sections of the paper derive the spectral algorithm for learning of L-PCFGs. The structure of these sections is as follows:

- Section 6 introduces a *tensor form* of the inside-outside algorithms for L-PCFGs. This is analogous to the matrix form for hidden Markov models (see Jaeger 2000, and in particular Lemma 1 of Hsu et al. 2009), and is also related to the use of tensors in spectral algorithms for directed graphical models (Parikh et al., 2011).
- Section 7.2 derives an *observable form* for the tensors required by algorithms of Section 6. The implication of this result is that the required tensors can be estimated directly from training data consisting of skeletal trees.
- Section 8 gives the algorithm for estimation of the tensors from a training sample, and gives a PAC-style generalization bound for the approach.

6. Tensor Form of the Inside-Outside Algorithm

This section first gives a tensor form of the inside-outside algorithms for L-PCFGs, then give an illustrative example.

6.1 The Tensor-Form Algorithms

Recall the two calculations for L-PCFGs introduced in Section 4.3:

Inputs: Sentence $x_1 \dots x_N$, L-PCFG $(\mathcal{N}, \mathcal{I}, \mathcal{P}, m, n, t, q, \pi)$, with parameters

- $t(a \to b\ c, h_2, h_3 | h_1, a)$ for all $a \to b\ c \in \mathcal{R}$, $h_1, h_2, h_3 \in [m]$.
- $q(a \to x | h, a)$ for all $a \in \mathcal{P}, x \in [n], h \in [m]$
- $\pi(a, h)$ for all $a \in \mathcal{I}$, $h \in [m]$.

Data structures:

- Each $\bar{\alpha}^{a,i,j} \in \mathbb{R}^{1 \times m}$ for $a \in \mathcal{N}$, $1 \leqslant i \leqslant j \leqslant N$ is a row vector of inside terms.
- Each $\bar{\beta}^{a,i,j} \in \mathbb{R}^{m \times 1}$ for $a \in \mathcal{N}$, $1 \leqslant i \leqslant j \leqslant N$ is a column vector of outside terms.
- Each $\bar{\mu}(a, i, j) \in \mathbb{R}$ for $a \in \mathcal{N}$, $1 \leqslant i \leqslant j \leqslant N$ is a marginal probability.

Algorithm:

(Inside base case) $\forall a \in \mathcal{P}, i \in [N], h \in [m]$ $\bar{\alpha}_h^{a,i,i} = q(a \to x_i | h, a)$

(Inside recursion) $\forall a \in \mathcal{I}, 1 \leqslant i < j \leqslant N, h \in [m]$

$$\bar{\alpha}_h^{a,i,j} = \sum_{k=i}^{j-1} \sum_{a \to b\ c} \sum_{h_2 \in [m]} \sum_{h_3 \in [m]} t(a \to b\ c, h_2, h_3 | h, a) \times \bar{\alpha}_{h_2}^{b,i,k} \times \bar{\alpha}_{h_3}^{c,k+1,j}$$

(Outside base case) $\forall a \in \mathcal{I}, h \in [m]$ $\bar{\beta}_h^{a,1,n} = \pi(a, h)$

(Outside recursion) $\forall a \in \mathcal{N}, 1 \leqslant i \leqslant j \leqslant N, h \in [m]$

$$\bar{\beta}_h^{a,i,j} = \sum_{k=1}^{i-1} \sum_{b \to c\ a} \sum_{h_2 \in [m]} \sum_{h_3 \in [m]} t(b \to c\ a, h_3, h | h_2, b) \times \bar{\beta}_{h_2}^{b,k,j} \times \bar{\alpha}_{h_3}^{c,k,i-1}$$

$$+ \sum_{k=j+1}^{N} \sum_{b \to a\ c} \sum_{h_2 \in [m]} \sum_{h_3 \in [m]} t(b \to a\ c, h, h_3 | h_2, b) \times \bar{\beta}_{h_2}^{b,i,k} \times \bar{\alpha}_{h_3}^{c,j+1,k}$$

(Marginals) $\forall a \in \mathcal{N}, 1 \leqslant i \leqslant j \leqslant N$,

$$\bar{\mu}(a, i, j) = \bar{\alpha}^{a,i,j} \bar{\beta}^{a,i,j} = \sum_{h \in [m]} \bar{\alpha}_h^{a,i,j} \bar{\beta}_h^{a,i,j}$$

Figure 3: The conventional form of the inside-outside algorithm, for calculation of marginal terms $\bar{\mu}(a, i, j)$.

Inputs: s-tree $r_1 \dots r_N$, L-PCFG $(\mathcal{N}, \mathcal{I}, \mathcal{P}, m, n)$, parameters

- $C^{a \to b\ c} \in \mathbb{R}^{(m \times m \times m)}$ for all $a \to b\ c \in \mathcal{R}$
- $c^{\infty}_{a \to x} \in \mathbb{R}^{(1 \times m)}$ for all $a \in \mathcal{P}, x \in [n]$
- $c^{1}_{a} \in \mathbb{R}^{(m \times 1)}$ for all $a \in \mathcal{I}$.

Algorithm: (calculate the f^i terms bottom-up in the tree)

- For all $i \in [N]$ such that $a_i \in \mathcal{P}$, $f^i = c^{\infty}_{r_i}$
- For all $i \in [N]$ such that $a_i \in \mathcal{I}$, $f^i = C^{r_i}(f^{\beta}, f^{\gamma})$ where β is the index of the left child of node i in the tree, and γ is the index of the right child.

Return: $f^1 c^1_{a_1} = p(r_1 \dots r_N)$

Figure 4: The tensor form for calculation of $p(r_1 \dots r_N)$.

1. For a given s-tree $r_1 \dots r_N$, calculate $p(r_1 \dots r_N)$.
2. For a given input sentence $x = x_1 \dots x_N$, calculate the marginal probabilities

$$\mu(a, i, j) = \sum_{\tau \in \mathcal{T}(x) : (a,i,j) \in \tau} p(\tau)$$

for each non-terminal $a \in \mathcal{N}$, for each (i, j) such that $1 \leqslant i \leqslant j \leqslant N$, where $\mathcal{T}(x)$ denotes the set of all possible s-trees for the sentence x, and we write $(a, i, j) \in \tau$ if non-terminal a spans words $x_i \dots x_j$ in the parse tree τ.

The tensor form of the inside-outside algorithms for these two problems are shown in Figures 4 and 5. Each algorithm takes the following inputs:

1. A tensor $C^{a \to b\ c} \in \mathbb{R}^{(m \times m \times m)}$ for each rule $a \to b\ c$.
2. A vector $c^{\infty}_{a \to x} \in \mathbb{R}^{(1 \times m)}$ for each rule $a \to x$.
3. A vector $c^{1}_{a} \in \mathbb{R}^{(m \times 1)}$ for each $a \in \mathcal{I}$.

The following theorem gives conditions under which the algorithms are correct:

Theorem 3 *Assume that we have an L-PCFG with parameters $q_{a \to x}$, $T^{a \to b\ c}$, π^a, and that there exist matrices $G^a \in \mathbb{R}^{(m \times m)}$ for all $a \in \mathcal{N}$ such that each G^a is invertible, and such that:*

1. *For all rules $a \to b\ c$, $C^{a \to b\ c}(y^1, y^2) = \left(T^{a \to b\ c}(y^1 G^b, y^2 G^c)\right)(G^a)^{-1}$.*
2. *For all rules $a \to x$, $c^{\infty}_{a \to x} = q_{a \to x}(G^a)^{-1}$.*
3. *For all $a \in \mathcal{I}$, $c^{1}_{a} = G^a \pi^a$.*

Then: 1) The algorithm in Figure 4 correctly computes $p(r_1 \dots r_N)$ under the L-PCFG. 2) The algorithm in Figure 5 correctly computes the marginals $\mu(a, i, j)$ under the L-PCFG.

Proof: see Section A.1. The next section (Section 6.2) gives an example that illustrates the basic intuition behind the proof. ■

Remark 4 *It is easily verified (see also the example in Section 6.2), that if the inputs to the tensor-form algorithms are of the following form (equivalently, the matrices G^a for all a are equal to the identity matrix):*

1. *For all rules $a \rightarrow b\ c$, $C^{a \rightarrow b\ c}(y^1, y^2) = T^{a \rightarrow b\ c}(y^1, y^2)$.*
2. *For all rules $a \rightarrow x$, $c^{\infty}_{a \rightarrow x} = q_{a \rightarrow x}$.*
3. *For all $a \in \mathcal{I}$, $c^1_a = \pi^a$.*

then the algorithms in Figures 4 and 5 are identical to the algorithms in Figures 2 and 3 respectively. More precisely, we have the identities

$$b^i_h = f^i_h$$

for the quantities in Figures 2 and 4, and

$$\bar{\alpha}^{a,i,j}_h = \alpha^{a,i,j}_h$$
$$\bar{\beta}^{a,i,j}_h = \beta^{a,i,j}_h$$

for the quantities in Figures 3 and 5.

The theorem shows, however, that it is sufficient[2] to have parameters that are equal to $T^{a \rightarrow b\ c}$, $q_{a \rightarrow x}$ and π^a up to linear transforms defined by the matrices G^a for all non-terminals a. The linear transformations add an extra degree of freedom that is crucial in what follows in this paper: in the next section, on observable representations, we show that it is possible to directly estimate values for $C^{a \rightarrow b\ c}$, $c^{\infty}_{a \rightarrow x}$ and c^1_a that satisfy the conditions of the theorem, but where the matrices G^a are not the identity matrix.

The key step in the proof of the theorem (see Section A.1) is to show that under the assumptions of the theorem we have the identities

$$f^i = b^i (G^a)^{-1}$$

for Figures 2 and 4, and

$$\alpha^{a,i,j} = \bar{\alpha}^{a,i,j} (G^a)^{-1}$$
$$\beta^{a,i,j} = G^a \bar{\beta}^{a,i,j}$$

for Figures 3 and 5. Thus the quantities calculated by the tensor-form algorithms are equivalent to the quantities calculated by the conventional algorithms, up to linear transforms. The linear transforms and their inverses cancel in useful ways: for example in the output from Figure 4 we have

$$\mu(a, i, j) = \alpha^{a,i,j} \beta^{a,i,j} = \bar{\alpha}^{a,i,j} (G^a)^{-1} G^a \bar{\beta}^{a,i,j} = \sum_h \bar{\alpha}^{a,i,j}_h \bar{\beta}^{a,i,j}_h,$$

showing that the marginals calculated by the conventional and tensor-form algorithms are identical.

2. Assuming that the goal is to calculate $p(r_1 \dots r_N)$ for any skeletal tree, or marginal terms $\mu(a, i, j)$.

Inputs: Sentence $x_1 \dots x_N$, L-PCFG $(\mathcal{N}, \mathcal{I}, \mathcal{P}, m, n)$, parameters $C^{a \to b\ c} \in \mathbb{R}^{(m \times m \times m)}$ for all $a \to b\ c \in \mathcal{R}$, $c^{\infty}_{a \to x} \in \mathbb{R}^{(1 \times m)}$ for all $a \in \mathcal{P}, x \in [n]$, $c^{1}_{a} \in \mathbb{R}^{(m \times 1)}$ for all $a \in \mathcal{I}$.
Data structures:

- Each $\alpha^{a,i,j} \in \mathbb{R}^{1 \times m}$ for $a \in \mathcal{N}$, $1 \leqslant i \leqslant j \leqslant N$ is a row vector of inside terms.
- Each $\beta^{a,i,j} \in \mathbb{R}^{m \times 1}$ for $a \in \mathcal{N}$, $1 \leqslant i \leqslant j \leqslant N$ is a column vector of outside terms.
- Each $\mu(a,i,j) \in \mathbb{R}$ for $a \in \mathcal{N}$, $1 \leqslant i \leqslant j \leqslant N$ is a marginal probability.

Algorithm:
(Inside base case) $\forall a \in \mathcal{P}, i \in [N]$, $\alpha^{a,i,i} = c^{\infty}_{a \to x_i}$
(Inside recursion) $\forall a \in \mathcal{I}, 1 \leqslant i < j \leqslant N$,

$$\alpha^{a,i,j} = \sum_{k=i}^{j-1} \sum_{a \to b\ c} C^{a \to b\ c}(\alpha^{b,i,k}, \alpha^{c,k+1,j})$$

(Outside base case) $\forall a \in \mathcal{I}$, $\beta^{a,1,n} = c^{1}_{a}$
(Outside recursion) $\forall a \in \mathcal{N}, 1 \leqslant i \leqslant j \leqslant N$,

$$\beta^{a,i,j} = \sum_{k=1}^{i-1} \sum_{b \to c\ a} C^{b \to c\ a}_{(1,2)}(\beta^{b,k,j}, \alpha^{c,k,i-1}) + \sum_{k=j+1}^{N} \sum_{b \to a\ c} C^{b \to a\ c}_{(1,3)}(\beta^{b,i,k}, \alpha^{c,j+1,k})$$

(Marginals) $\forall a \in \mathcal{N}, 1 \leqslant i \leqslant j \leqslant N$,

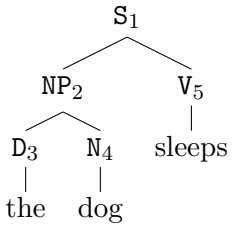
$$\mu(a,i,j) = \alpha^{a,i,j}\beta^{a,i,j} = \sum_{h \in [m]} \alpha^{a,i,j}_{h} \beta^{a,i,j}_{h}$$

Figure 5: The tensor form of the inside-outside algorithm, for calculation of marginal terms $\mu(a,i,j)$.

6.2 An Example

In the remainder of this section we give an example that illustrates how the algorithm in Figure 4 is correct, and gives the basic intuition behind the proof in Section A.1. While we concentrate on the algorithm in Figure 4, the intuition behind the algorithm in Figure 5 is very similar.

Consider the skeletal tree in Figure 6. We will demonstrate how the algorithm in Figure 4, under the assumptions in the theorem, correctly calculates the probability of this tree. In brief, the argument involves the following steps:



$r_1 = \texttt{S} \rightarrow \texttt{NP V}$
$r_2 = \texttt{NP} \rightarrow \texttt{D N}$
$r_3 = \texttt{D} \rightarrow \texttt{the}$
$r_4 = \texttt{N} \rightarrow \texttt{dog}$
$r_5 = \texttt{V} \rightarrow \texttt{sleeps}$

Figure 6: An s-tree, and its sequence of rules. (For convenience we have numbered the nodes in the tree.)

1. We first show that the algorithm in Figure 4, when run on the tree in Figure 6, calculates the probability of the tree as

$$C^{S \rightarrow NP\ V}(C^{NP \rightarrow D\ N}(c^{\infty}_{D \rightarrow the}, c^{\infty}_{N \rightarrow dog}), c^{\infty}_{V \rightarrow sleeps}) c^{1}_{S}.$$

   Note that this expression mirrors the structure of the tree, with $c^{\infty}_{a \rightarrow x}$ terms for the leaves, $C^{a \rightarrow b\ c}$ terms for each rule production $a \rightarrow b\ c$ in the tree, and a c^{1}_{S} term for the root.

2. We then show that under the assumptions in the theorem, the following identity holds:

$$\begin{aligned} & C^{S \rightarrow NP\ V}(C^{NP \rightarrow D\ N}(c^{\infty}_{D \rightarrow the}, c^{\infty}_{N \rightarrow dog}), c^{\infty}_{V \rightarrow sleeps}) c^{1}_{S}. \\ = \quad & T^{S \rightarrow NP\ V}(T^{NP \rightarrow D\ N}(q_{D \rightarrow the}, q_{N \rightarrow dog}), q_{V \rightarrow sleeps}) \pi^{S} \end{aligned} \tag{3}$$

   This follows because the G^a and $(G^a)^{-1}$ terms for the various non-terminals in the tree cancel. Note that the expression in Eq. 3 again follows the structure of the tree, but with $q_{a \rightarrow x}$ terms for the leaves, $T^{a \rightarrow b\ c}$ terms for each rule production $a \rightarrow b\ c$ in the tree, and a π^{S} term for the root.

3. Finally, we show that the expression in Eq. 3 implements the conventional dynamic-programming method for calculation of the tree probability, as described in Eqs. 11–13 below.

We now go over these three points in detail. The algorithm in Figure 4 calculates the following terms (each f^i is an m-dimensional row vector):

$$\begin{aligned} f^3 &= c^{\infty}_{D \rightarrow the} \\ f^4 &= c^{\infty}_{N \rightarrow dog} \\ f^5 &= c^{\infty}_{V \rightarrow sleeps} \\ f^2 &= C^{NP \rightarrow D\ N}(f^3, f^4) \\ f^1 &= C^{S \rightarrow NP\ V}(f^2, f^5) \end{aligned}$$

The final quantity returned by the algorithm is

$$f^1 c_S^1 = \sum_h f_h^1 [c_S^1]_h.$$

Combining the definitions above, it can be seen that

$$f^1 c_S^1 = C^{S \to NP\ V}(C^{NP \to D\ N}(c^\infty_{D \to the}, c^\infty_{N \to dog}), c^\infty_{V \to sleeps}) c_S^1,$$

demonstrating that point 1 above holds.

Next, given the assumptions in the theorem, we show point 2, that is, that

$$\begin{aligned} & C^{S \to NP\ V}(C^{NP \to D\ N}(c^\infty_{D \to the}, c^\infty_{N \to dog}), c^\infty_{V \to sleeps}) c_S^1 \\ = \ & T^{S \to NP\ V}(T^{NP \to D\ N}(q_{D \to the}, q_{N \to dog}), q_{V \to sleeps}) \pi^S. \end{aligned} \tag{4}$$

This follows because the G^a and $(G^a)^{-1}$ terms in the theorem cancel. More specifically, we have

$$\begin{aligned} f^3 &= c^\infty_{D \to the} = q_{D \to the} (G^D)^{-1} && (5) \\ f^4 &= c^\infty_{N \to dog} = q_{N \to dog} (G^N)^{-1} && (6) \\ f^5 &= c^\infty_{V \to sleeps} = q_{V \to sleeps} (G^V)^{-1} && (7) \\ f^2 &= C^{NP \to D\ N}(f^3, f^4) = T^{NP \to D\ N}(q_{D \to the}, q_{D \to dog})(G^{NP})^{-1} && (8) \\ f^1 &= C^{S \to NP\ V}(f^2, f^5) = T^{S \to NP\ V}(T^{NP \to D\ N}(q_{D \to the}, q_{N \to dog}), q_{V \to sleeps})(G^S)^{-1} && (9) \end{aligned}$$

Eqs. 5, 6, 7 follow by the assumptions in the theorem. Eq. 8 follows because by the assumptions in the theorem

$$C^{NP \to D\ N}(f^3, f^4) = T^{NP \to D\ N}(f^3 G^D, f^4 G^N)(G^{NP})^{-1}$$

hence

$$\begin{aligned} C^{NP \to D\ N}(f^3, f^4) &= T^{NP \to D\ N}(q_{D \to the}(G^D)^{-1} G^D, q_{N \to dog}(G^N)^{-1} G^N)(G^{NP})^{-1} \\ &= T^{NP \to D\ N}(q_{D \to the}, q_{N \to dog})(G^{NP})^{-1} \end{aligned}$$

Eq. 9 follows in a similar manner.

It follows by the assumption that $c_S^1 = G^S \pi^S$ that

$$\begin{aligned} & C^{S \to NP\ V}(C^{NP \to D\ N}(c^\infty_{D \to the}, c^\infty_{N \to dog}), c^\infty_{V \to sleeps}) c_S^1 \\ = \ & T^{S \to NP\ V}(T^{NP \to D\ N}(q_{D \to the}, q_{N \to dog}), q_{V \to sleeps})(G^S)^{-1} G^S \pi^S \\ = \ & T^{S \to NP\ V}(T^{NP \to D\ N}(q_{D \to the}, q_{N \to dog}), q_{V \to sleeps}) \pi^S \end{aligned} \tag{10}$$

The final step (point 3) is to show that the expression in Eq. 10 correctly calculates the probability of the example tree. First consider the term $T^{NP \to D\ N}(q_{D \to the}, q_{N \to dog})$—this

is an m-dimensional row vector, call this b^2. By the definition of the tensor $T^{NP \to D\ N}$, we have

$$\begin{aligned} b^2_h &= \left[T^{NP \to D\ N}(q_{D \to the}, q_{N \to dog})\right]_h \\ &= \sum_{h_2, h_3} t(NP \to D\ N, h_2, h_3 | h, NP) \times q(D \to the | h_2, D) \times q(N \to dog | h_3, N) \end{aligned} \quad (11)$$

By a similar calculation, $T^{S \to NP\ V}(T^{NP \to D\ N}(q_{D \to the}, q_{N \to dog}), q_{V \to sleeps})$—call this vector b^1—is

$$b^1_h = \sum_{h_2, h_3} t(S \to NP\ V, h_2, h_3 | h, S) \times b^2_{h_2} \times q(V \to sleeps | h_3, V) \quad (12)$$

Finally, the probability of the full tree is calculated as

$$\sum_h b^1_h \pi^S_h. \quad (13)$$

It can be seen that the expression in Eq. 4 implements the calculations in Eqs. 11, 12 and 13, which are precisely the calculations used in the conventional dynamic programming algorithm for calculation of the probability of the tree.

7. Estimating the Tensor Model

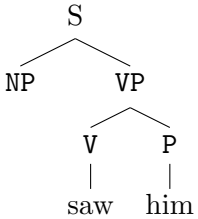
A crucial result is that it is possible to directly estimate parameters $C^{a \to b\ c}$, $c^{\infty}_{a \to x}$ and c^1_a that satisfy the conditions in Theorem 3, from a training sample consisting of s-trees (i.e., trees where hidden variables are unobserved). We first describe random variables underlying the approach, then describe observable representations based on these random variables.

7.1 Random Variables Underlying the Approach

Each s-tree with N rules $r_1 \ldots r_N$ has N nodes. We will use the s-tree in Figure 1 as a running example.

Each node has an associated rule: for example, node 2 in the tree in Figure 1 has the rule `NP` $\to$ `D N`. If the rule at a node is of the form $a \to b\ c$, then there are left and right *inside trees* below the left child and right child of the rule. For example, for node 2 we have a left inside tree rooted at node 3, and a right inside tree rooted at node 4 (in this case the left and right inside trees both contain only a single rule production, of the form $a \to x$; however in the general case they might be arbitrary subtrees).

In addition, each node has an *outside* tree. For node 2, the outside tree is



The outside tree contains everything in the s-tree $r_1 \ldots r_N$, excluding the subtree below node i.

Our random variables are defined as follows. First, we select a random internal node, from a random tree, as follows:

- Sample a full tree $r_1 \ldots r_N, h_1 \ldots h_N$ from the PMF $p(r_1 \ldots r_N, h_1 \ldots h_N)$.
- Choose a node i uniformly at random from $[N]$.

If the rule r_i for the node i is of the form $a \rightarrow b\ c$, we define random variables as follows:

- R_1 is equal to the rule r_i (e.g., $\texttt{NP} \rightarrow \texttt{D}\ \texttt{N}$).
- T_1 is the inside tree rooted at node i. T_2 is the inside tree rooted at the left child of node i, and T_3 is the inside tree rooted at the right child of node i.
- H_1, H_2, H_3 are the hidden variables associated with node i, the left child of node i, and the right child of node i respectively.
- A_1, A_2, A_3 are the labels for node i, the left child of node i, and the right child of node i respectively. (e.g., $A_1 = \texttt{NP}$, $A_2 = \texttt{D}$, $A_3 = \texttt{N}$.)
- O is the outside tree at node i.
- B is equal to 1 if node i is at the root of the tree (i.e., $i = 1$), 0 otherwise.

If the rule r_i for the selected node i is of the form $a \rightarrow x$, we have random variables R_1, T_1, H_1,
A_1, O, B as defined above, but H_2, H_3, T_2, T_3, A_2, and A_3 are not defined.

We assume a function ψ that maps outside trees o to feature vectors $\psi(o) \in \mathbb{R}^{d'}$. For example, the feature vector might track the rule directly above the node in question, the word following the node in question, and so on. We also assume a function ϕ that maps inside trees t to feature vectors $\phi(t) \in \mathbb{R}^d$. As one example, the function ϕ might be an indicator function tracking the rule production at the root of the inside tree. Later we give formal criteria for what makes good definitions of $\psi(o)$ and $\phi(t)$. One requirement is that $d' \geqslant m$ and $d \geqslant m$.

In tandem with these definitions, we assume projection matrices $U^a \in \mathbb{R}^{(d \times m)}$ and $V^a \in \mathbb{R}^{(d' \times m)}$ for all $a \in \mathcal{N}$. We then define additional random variables Y_1, Y_2, Y_3, Z as

$$Y_1 = (U^{a_1})^\top \phi(T_1) \quad Z = (V^{a_1})^\top \psi(O)$$
$$Y_2 = (U^{a_2})^\top \phi(T_2) \quad Y_3 = (U^{a_3})^\top \phi(T_3)$$

where a_i is the value of the random variable A_i. Note that Y_1, Y_2, Y_3, Z are all in $\mathbb{R}^m$.

7.2 Observable Representations

Given the definitions in the previous section, our representation is based on the following matrix, tensor and vector quantities, defined for all $a \in \mathcal{N}$, for all rules of the form $a \rightarrow b\ c$, and for all rules of the form $a \rightarrow x$ respectively:

$$\begin{aligned}
\Sigma^a &= \mathbf{E}[Y_1 Z^\top | A_1 = a], \\
D^{a \rightarrow b\ c} &= \mathbf{E}\left[[\![R_1 = a \rightarrow b\ c]\!] Z Y_2^\top Y_3^\top | A_1 = a\right], \\
d^\infty_{a \rightarrow x} &= \mathbf{E}\left[[\![R_1 = a \rightarrow x]\!] Z^\top | A_1 = a\right].
\end{aligned}$$

Assuming access to functions ϕ and ψ, and projection matrices U^a and V^a, these quantities can be estimated directly from training data consisting of a set of s-trees (see Section 8).

Our observable representation then consists of:

$$C^{a\to b\ c}(y^1, y^2) = D^{a\to b\ c}(y^1, y^2)(\Sigma^a)^{-1}, \quad (14)$$

$$c^{\infty}_{a\to x} = d^{\infty}_{a\to x}(\Sigma^a)^{-1}, \quad (15)$$

$$c^1_a = \mathbf{E}\left[[\![A_1 = a]\!] Y_1 | B = 1\right]. \quad (16)$$

We next introduce conditions under which these quantities satisfy the conditions in Theorem 3.

The following definition will be important:

Definition 5 *For all $a \in \mathcal{N}$, we define the matrices $I^a \in \mathbb{R}^{(d\times m)}$ and $J^a \in \mathbb{R}^{(d'\times m)}$ as*

$$[I^a]_{i,h} = \mathbf{E}[\phi_i(T_1) \mid H_1 = h, A_1 = a],$$

$$[J^a]_{i,h} = \mathbf{E}[\psi_i(O) \mid H_1 = h, A_1 = a].$$

In addition, for any $a \in \mathcal{N}$, we use $\gamma^a \in \mathbb{R}^m$ to denote the vector with $\gamma^a_h = P(H_1 = h | A_1 = a)$.

The correctness of the representation will rely on the following conditions being satisfied (these are parallel to conditions 1 and 2 in Hsu et al. (2009)):

Condition 1 *$\forall a \in \mathcal{N}$, the matrices I^a and J^a are of full rank (i.e., they have rank m). For all $a \in \mathcal{N}$, for all $h \in [m]$, $\gamma^a_h > 0$.*

Condition 2 *$\forall a \in \mathcal{N}$, the matrices $U^a \in \mathbb{R}^{(d\times m)}$ and $V^a \in \mathbb{R}^{(d'\times m)}$ are such that the matrices $G^a = (U^a)^\top I^a$ and $K^a = (V^a)^\top J^a$ are invertible.*

We can now state the following theorem:

Theorem 6 *Assume conditions 1 and 2 are satisfied. For all $a \in \mathcal{N}$, define $G^a = (U^a)^\top I^a$. Then under the definitions in Eqs. 14-16:*

1. *For all rules $a \to b\ c$, $C^{a\to b\ c}(y^1, y^2) = \left(T^{a\to b\ c}(y^1 G^b, y^2 G^c)\right)(G^a)^{-1}$*
2. *For all rules $a \to x$, $c^{\infty}_{a\to x} = q_{a\to x}(G^a)^{-1}$.*
3. *For all $a \in \mathcal{N}$, $c^1_a = G^a \pi^a$*

Proof: The following identities hold (see Section A.2):

$$D^{a\to b\ c}(y^1, y^2) = \left(T^{a\to b\ c}(y^1 G^b, y^2 G^c)\right) \mathrm{diag}(\gamma^a)(K^a)^\top \quad (17)$$

$$d^{\infty}_{a\to x} = q_{a\to x}\mathrm{diag}(\gamma^a)(K^a)^\top \quad (18)$$

$$\Sigma^a = G^a \mathrm{diag}(\gamma^a)(K^a)^\top \quad (19)$$

$$c^1_a = G^a \pi^a \quad (20)$$

Under conditions 1 and 2, Σ^a is invertible, and $(\Sigma^a)^{-1} = ((K^a)^\top)^{-1}(\text{diag}(\gamma^a))^{-1}(G^a)^{-1}$. The identities in the theorem follow immediately. ■

This theorem leads directly to the spectral learning algorithm, which we describe in the next section. We give a sketch of the approach here. Assume that we have a training set consisting of skeletal trees (no latent variables are observed) generated from some underlying L-PCFG. Assume in addition that we have definitions of ϕ, ψ, U^a and V^a such that conditions 1 and 2 are satisfied for the L-PCFG. Then it is straightforward to use the training examples to derive i.i.d. samples from the joint distribution over the random variables $(A_1, R_1, Y_1, Y_2, Y_3, Z, B)$ used in the definitions in Eqs. 14–16. These samples can be used to estimate the quantities in Eqs. 14–16; the estimated quantities $\hat{C}^{a\to b\,c}$, $\hat{c}^{\infty}_{a\to x}$ and $\hat{c}^1_a$ can then be used as inputs to the algorithms in Figures 4 and 5. By standard arguments, the estimates $\hat{C}^{a\to b\,c}$, $\hat{c}^{\infty}_{a\to x}$ and $\hat{c}^1_a$ will converge to the values in Eqs. 14–16.

The following lemma justifies the use of an SVD calculation as one method for finding values for U^a and V^a that satisfy condition 2, assuming that condition 1 holds:

Lemma 7 *Assume that condition 1 holds, and for all $a \in \mathcal{N}$ define*

$$\Omega^a = \mathbf{E}[\phi(T_1)\,(\psi(O))^\top \,|A_1 = a] \tag{21}$$

Then if U^a is a matrix of the m left singular vectors of Ω^a corresponding to non-zero singular values, and V^a is a matrix of the m right singular vectors of Ω^a corresponding to non-zero singular values, then condition 2 is satisfied.

Proof sketch: It can be shown that $\Omega^a = I^a \text{diag}(\gamma^a)(J^a)^\top$. The remainder is similar to the proof of lemma 2 in Hsu et al. (2009). ■

The matrices Ω^a can be estimated directly from a training set consisting of s-trees, assuming that we have access to the functions ϕ and ψ. Similar arguments to those of Hsu et al. (2009) can be used to show that with a sufficient number of samples, the resulting estimates of U^a and V^a satisfy condition 2 with high probability.

8. Deriving Empirical Estimates

Figure 7 shows an algorithm that derives estimates of the quantities in Eqs. 14, 15, and 16. As input, the algorithm takes a sequence of tuples $(r^{(i,1)}, t^{(i,1)}, t^{(i,2)}, t^{(i,3)}, o^{(i)}, b^{(i)})$ for $i \in [M]$.

These tuples can be derived from a training set consisting of s-trees $\tau_1 \ldots \tau_M$ as follows:

• $\forall i \in [M]$, choose a single node j_i uniformly at random from the nodes in τ_i. Define $r^{(i,1)}$ to be the rule at node j_i. $t^{(i,1)}$ is the inside tree rooted at node j_i. If $r^{(i,1)}$ is of the form $a \to b\ c$, then $t^{(i,2)}$ is the inside tree under the left child of node j_i, and $t^{(i,3)}$ is the inside tree under the right child of node j_i. If $r^{(i,1)}$ is of the form $a \to x$, then $t^{(i,2)} = t^{(i,3)} = \texttt{NULL}$. $o^{(i)}$ is the outside tree at node j_i. $b^{(i)}$ is 1 if node j_i is at the root of the tree, 0 otherwise.

Under this process, assuming that the s-trees $\tau_1 \ldots \tau_M$ are i.i.d. draws from the distribution $p(\tau)$ over s-trees under an L-PCFG, the tuples $(r^{(i,1)}, t^{(i,1)}, t^{(i,2)}, t^{(i,3)}, o^{(i)}, b^{(i)})$ are i.i.d. draws from the joint distribution over the random variables R_1, T_1, T_2, T_3, O, B defined in the previous section.

The algorithm first computes estimates of the projection matrices U^a and V^a: following Lemma 7, this is done by first deriving estimates of Ω^a, and then taking SVDs of each Ω^a.

The matrices are then used to project inside and outside trees $t^{(i,1)}, t^{(i,2)}, t^{(i,3)}, o^{(i)}$ down to m-dimensional vectors $y^{(i,1)}, y^{(i,2)}, y^{(i,3)}, z^{(i)}$; these vectors are used to derive the estimates of $C^{a \to b\ c}$, $c^{\infty}_{a \to x}$, and c^{1}_{a}. For example, the quantities

$$\begin{aligned} D^{a \to b\ c} &= \mathbf{E}\left[[\![R_1 = a \to b\ c]\!] Z Y_2^\top Y_3^\top | A_1 = a\right] \\ d^{\infty}_{a \to x} &= \mathbf{E}\left[[\![R_1 = a \to x]\!] Z^\top | A_1 = a\right] \end{aligned}$$

can be estimated as

$$\hat{D}^{a \to b\ c} = \delta_a \times \sum_{i=1}^{M} [\![r^{(i,1)} = a \to b\ c]\!] z^{(i)} (y^{(i,2)})^\top (y^{(i,3)})^\top$$

$$\hat{d}^{\infty}_{a \to x} = \delta_a \times \sum_{i=1}^{M} [\![r^{(i,1)} = a \to x]\!] (z^{(i)})^\top$$

where $\delta_a = 1 / \sum_{i=1}^{M} [\![a_i = a]\!]$, and we can then set

$$\hat{C}^{a \to b\ c}(y^1, y^2) = \hat{D}^{a \to b\ c}(y^1, y^2)(\hat{\Sigma}^a)^{-1}$$

$$\hat{c}^{\infty}_{a \to x} = \hat{d}^{\infty}_{a \to x}(\hat{\Sigma}^a)^{-1}.$$

We now state a PAC-style theorem for the learning algorithm. First, we give the following assumptions and definitions:

- We have an L-PCFG $(\mathcal{N}, \mathcal{I}, \mathcal{P}, m, n, t, q, \pi)$. The samples used in Figures 7 and 8 are i.i.d. samples from the L-PCFG (for simplicity of analysis we assume that the two algorithms use independent sets of M samples each: see above for how to draw i.i.d. samples from the L-PCFG).
- We have functions $\phi(t) \in \mathbb{R}^d$ and $\psi(o) \in \mathbb{R}^{d'}$ that map inside and outside trees respectively to feature vectors. We will assume without loss of generality that for all inside trees $||\phi(t)||_2 \leqslant 1$, and for all outside trees $||\psi(o)||_2 \leqslant 1$.
- See Section 7.2 for a definition of the random variables

  $$(R_1, T_1, T_2, T_3, A_1, A_2, A_3, H_1, H_2, H_3, O, B),$$

  and the joint distribution over them.
- For all $a \in \mathcal{N}$ define

  $$\Omega^a = \mathbf{E}[\phi(T_1)(\psi(O))^\top | A_1 = a]$$

  and define $I^a \in \mathbb{R}^{d \times m}$ to be the matrix with entries

  $$[I^a]_{i,h} = \mathbf{E}[\phi_i(T_1) | A_1 = a, H_1 = h]$$

- Define

  $$\sigma = \min_a \sigma_m(\Omega^a)$$

  and

  $$\xi = \min_a \sigma_m(I^a)$$

  where $\sigma_m(A)$ is the m'th largest singular value of the matrix A.

- Define
$$\gamma = \min_{a,b,c \in \mathcal{N}, h_1, h_2, h_3 \in [m]} t(a \rightarrow b\ c, h_2, h_3 | a, h_1)$$
- Define $\mathcal{T}(a, N)$ to be the set of of all skeletal trees with N binary rules (hence $2N+1$ rules in total), with non-terminal a at the root of the tree.

The following theorem gives a bound on the sample complexity of the algorithm:

Theorem 8 *There exist constants C_1, C_2, C_3, C_4, C_5 such that the following holds. Pick any $\epsilon > 0$, any value for δ such that $0 < \delta < 1$, and any integer N such that $N \geqslant 1$. Define $L = \log \frac{2|\mathcal{N}|+1}{\delta}$. Assume that the parameters $\hat{C}^{a \rightarrow b\ c}$, $\hat{c}^{\infty}_{a \rightarrow x}$ and $\hat{c}^{1}_{a}$ are output from the algorithm in Figure 7, with values for N_a, M_a and R such that*

$$\forall a \in \mathcal{I}, N_a \geqslant \frac{C_1 L N^2 m^2}{\gamma^2 \epsilon^2 \xi^4 \sigma^4} \quad \forall a \in \mathcal{P}, N_a \geqslant \frac{C_2 L N^2 m^2 n}{\epsilon^2 \sigma^4}$$

$$\forall a \in \mathcal{I}, M_a \geqslant \frac{C_3 L N^2 m^2}{\gamma^2 \epsilon^2 \xi^4 \sigma^2} \quad \forall a \in \mathcal{P}, M_a \geqslant \frac{C_4 L N^2 m^2}{\epsilon^2 \sigma^2}$$

$$R \geqslant \frac{C_5 L N^2 m^3}{\epsilon^2 \sigma^2}$$

It follows that with probability at least $1 - \delta$, for all $a \in \mathcal{N}$,

$$\sum_{t \in \mathcal{T}(a,N)} |\hat{p}(t) - p(t)| \leqslant \epsilon$$

, where $\hat{p}(t)$ is the output from the algorithm in Figure 4 with parameters $\hat{C}^{a \rightarrow b\ c}$, $\hat{c}^{\infty}_{a \rightarrow x}$ and $\hat{c}^{1}_{a}$, and $p(t)$ is the probability of the skeletal tree under the L-PCFG.

See Appendix B for a proof.

The method described of selecting a single tuple $(r^{(i,1)}, t^{(i,1)}, t^{(i,2)}, t^{(i,3)}, o^{(i)}, b^{(i)})$ for each s-tree ensures that the samples are i.i.d., and simplifies the analysis underlying Theorem 8. In practice, an implementation should use all nodes in all trees in training data; by Rao-Blackwellization we know such an algorithm would be better than the one presented, but the analysis of how much better would be challenging (Bickel and Doksum, 2006; section 3.4.2). It would almost certainly lead to a faster rate of convergence of $\hat{p}$ to p.

9. Discussion

There are several applications of the method. The most obvious is parsing with L-PCFGs (Cohen et al., 2013).[3] The approach should be applicable in other cases where EM has traditionally been used, for example in semi-supervised learning. Latent-variable HMMs for sequence labeling can be derived as special case of our approach, by converting tagged sequences to right-branching skeletal trees (Stratos et al., 2013).

3. Parameters can be estimated using the algorithm in Figure 7; for a test sentence $x_1 \ldots x_N$ we can first use the algorithm in Figure 5 to calculate marginals $\mu(a, i, j)$, then use the algorithm of Goodman (1996) to find $\arg\max_{\tau \in \mathcal{T}(x)} \sum_{(a,i,j) \in \tau} \mu(a, i, j)$.

Inputs: Training examples $(r^{(i,1)}, t^{(i,1)}, t^{(i,2)}, t^{(i,3)}, o^{(i)}, b^{(i)})$ for $i \in \{1 \dots M\}$, where $r^{(i,1)}$ is a context free rule; $t^{(i,1)}$, $t^{(i,2)}$ and $t^{(i,3)}$ are inside trees; $o^{(i)}$ is an outside tree; and $b^{(i)} = 1$ if the rule is at the root of tree, 0 otherwise. A function ϕ that maps inside trees t to feature-vectors $\phi(t) \in \mathbb{R}^d$. A function ψ that maps outside trees o to feature-vectors $\psi(o) \in \mathbb{R}^{d'}$.

Definitions: For each $a \in \mathcal{N}$, define $N_a = \sum_{i=1}^{M} [\![a_i = a]\!]$. Define $R = \sum_{i=1}^{M} [\![b^{(i)} = 1]\!]$. (These definitions will be used in Theorem 8.)

Algorithm:
Define a_i to be the non-terminal on the left-hand side of rule $r^{(i,1)}$. If $r^{(i,1)}$ is of the form $a \to b\ c$, define b_i to be the non-terminal for the left-child of $r^{(i,1)}$, and c_i to be the non-terminal for the right-child.
(Step 0: Singular Value Decompositions)

- Use the algorithm in Figure 8 to calculate matrices $\hat{U}^a \in \mathbb{R}^{(d \times m)}$, $\hat{V}^a \in \mathbb{R}^{(d' \times m)}$ and $\hat{\Sigma}^a \in \mathbb{R}^{(m \times m)}$ for each $a \in \mathcal{N}$.

(Step 1: Projection)

- For all $i \in [M]$, compute $y^{(i,1)} = (\hat{U}^{a_i})^\top \phi(t^{(i,1)})$.
- For all $i \in [M]$ such that $r^{(i,1)}$ is of the form $a \to b\ c$, compute $y^{(i,2)} = (\hat{U}^{b_i})^\top \phi(t^{(i,2)})$ and $y^{(i,3)} = (\hat{U}^{c_i})^\top \phi(t^{(i,3)})$.
- For all $i \in [M]$, compute $z^{(i)} = (\hat{V}^{a_i})^\top \psi(o^{(i)})$.

(Step 2: Calculate Correlations)

- For each $a \in \mathcal{N}$, define $\delta_a = 1 / \sum_{i=1}^{M} [\![a_i = a]\!]$.
- For each rule $a \to b\ c$, compute

$$\hat{D}^{a \to b\ c} = \delta_a \times \sum_{i=1}^{M} [\![r^{(i,1)} = a \to b\ c]\!] z^{(i)} (y^{(i,2)})^\top (y^{(i,3)})^\top.$$

- For each rule $a \to x$, compute $\hat{d}^{\infty}_{a \to x} = \delta_a \times \sum_{i=1}^{M} [\![r^{(i,1)} = a \to x]\!] (z^{(i)})^\top$.

(Step 3: Compute Final Parameters)

- For all $a \to b\ c$, $\hat{C}^{a \to b\ c}(y^1, y^2) = \hat{D}^{a \to b\ c}(y^1, y^2)(\hat{\Sigma}^a)^{-1}$.
- For all $a \to x$, $\hat{c}^{\infty}_{a \to x} = \hat{d}^{\infty}_{a \to x}(\hat{\Sigma}^a)^{-1}$.
- For all $a \in \mathcal{I}$, $\hat{c}^1_a = \frac{\sum_{i=1}^{M} [\![a_i = a \text{ and } b^{(i)} = 1]\!] y^{(i,1)}}{\sum_{i=1}^{M} [\![b^{(i)} = 1]\!]}$.

Figure 7: The spectral learning algorithm.

Inputs: Identical to algorithm in Figure 7.
Definition: For each $a \in \mathcal{N}$, define $M_a = \sum_{i=1}^{M} [\![a_i = a]\!]$ (this definition will be used in Theorem 8).
Algorithm:
• For each $a \in \mathcal{N}$, compute $\hat{\Omega}^a \in \mathbb{R}^{(d \times d')}$ as

$$\hat{\Omega}^a = \frac{\sum_{i=1}^{M} [\![a_i = a]\!] \phi(t^{(i,1)})(\psi(o^{(i)}))^\top}{\sum_{i=1}^{M} [\![a_i = a]\!]}$$

and calculate a singular value decomposition of $\hat{\Omega}^a$.
• For each $a \in \mathcal{N}$, define $\hat{U}^a \in \mathbb{R}^{m \times d}$ to be a matrix of the left singular vectors of $\hat{\Omega}^a$ corresponding to the m largest singular values. Define $\hat{V}^a \in \mathbb{R}^{m \times d'}$ to be a matrix of the right singular vectors of $\hat{\Omega}^a$ corresponding to the m largest singular values. Define $\hat{\Sigma}^a = (\hat{U}^a)^\top \hat{\Omega}^a \hat{V}^a$.

Figure 8: Singular value decompositions.

In terms of efficiency, the first step of the algorithm in Figure 7 requires an SVD calculation: modern methods for calculating SVDs are very efficient (e.g., see Dhillon et al., 2011 and Tropp et al., 2009). The remaining steps of the algorithm require manipulation of tensors or vectors, and require $O(Mm^3)$ time.

The sample complexity of the method depends on the minimum singular values of Ω^a; these singular values are a measure of how well correlated ψ and ϕ are with the unobserved hidden variable H_1. Experimental work is required to find a good choice of values for ψ and ϕ for parsing.

For simplicity we have considered the case where each non-terminal has the same number, m, of possible hidden values. It is simple to generalize the algorithms to the case where the number of hidden values varies depending on the non-terminal; this is important in applications such as parsing.

Acknowledgements

The authors gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA, AFRL, or the US government. Shay Cohen was supported by the National Science Foundation under Grant #1136996 to the Computing Research Association for the CIFellows Project. Dean Foster was supported by National Science Foundation grant 1106743. This work also used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575.

Appendix A. Proofs of Theorems 1 and 2

This section gives proofs of Theorems 3 and 6.

A.1 Proof of Theorem 3

The key idea behind the proof of Theorem 3 is to show that the algorithms in Figures 4 and 5 compute the same quantities as the conventional version of the inside outside algorithms, as shown in Figures 2 and 3.

First, the following lemma leads directly to the correctness of the algorithm in Figure 4:

Lemma 9 *Assume that conditions 1-3 of Theorem 3 are satisfied, and that the input to the algorithm in Figure 4 is an s-tree $r_1 \dots r_N$. Define a_i for $i \in [N]$ to be the non-terminal on the left-hand-side of rule r_i. For all $i \in [N]$, define the row vector $b^i \in \mathbb{R}^{(1\times m)}$ to be the vector computed by the conventional inside-outside algorithm, as shown in Figure 2, on the s-tree $r_1 \dots r_N$. Define $f^i \in \mathbb{R}^{(1\times m)}$ to be the vector computed by the tensor-based inside-outside algorithm, as shown in Figure 4, on the s-tree $r_1 \dots r_N$.*

Then for all $i \in [N]$, $f^i = b^i (G^{(a_i)})^{-1}$. It follows immediately that

$$f^1 c^1_{a_1} = b^1 (G^{(a_1)})^{-1} G^{a_1} \pi_{a_1} = b^1 \pi_{a_1} = \sum_h b^1_h \pi(a, h).$$

Hence the output from the algorithms in Figures 2 and 4 is the same, and it follows that the tensor-based algorithm in Figure 4 is correct.

This lemma shows a direct link between the vectors f^i calculated in the algorithm, and the terms b^i_h, which are terms calculated by the conventional inside algorithm: each f^i is a linear transformation (through G^{a_i}) of the corresponding vector b^i.

Proof: The proof is by induction.

First consider the base case. For any leaf—i.e., for any i such that $a_i \in \mathcal{P}$—we have $b^i_h = q(r_i|h, a_i)$, and it is easily verified that $f^i = b^i (G^{(a_i)})^{-1}$.

The inductive case is as follows. For all $i \in [N]$ such that $a_i \in \mathcal{I}$, by the definition in the algorithm,

$$\begin{aligned} f^i &= C^{r_i}(f^\beta, f^\gamma) \\ &= \left(T^{r_i}(f^\beta G^{a_\beta}, f^\gamma G^{a_\gamma})\right)(G^{a_i})^{-1} \end{aligned}$$

Assuming by induction that $f^\beta = b^\beta (G^{(a_\beta)})^{-1}$ and $f^\gamma = b^\gamma (G^{(a_\gamma)})^{-1}$, this simplifies to

$$f^i = \left(T^{r_i}(b^\beta, b^\gamma)\right)(G^{a_i})^{-1}. \tag{22}$$

By the definition of the tensor T^{r_i},

$$\left[T^{r_i}(b^\beta, b^\gamma)\right]_h = \sum_{h_2 \in [m], h_3 \in [m]} t(r_i, h_2, h_3 | a_i, h) b^\beta_{h_2} b^\gamma_{h_3}$$

But by definition (see the algorithm in Figure 2),

$$b^i_h = \sum_{h_2 \in [m], h_3 \in [m]} t(r_i, h_2, h_3 | a_i, h) b^\beta_{h_2} b^\gamma_{h_3},$$

hence $b^i = T^{r_i}(b^\beta, b^\gamma)$ and the inductive case follows immediately from Eq. 22. ■

Next, we give a similar lemma, which implies the correctness of the algorithm in Figure 5:

Lemma 10 *Assume that conditions 1-3 of Theorem 3 are satisfied, and that the input to the algorithm in Figure 5 is a sentence $x_1 \dots x_N$. For any $a \in \mathcal{N}$, for any $1 \leqslant i \leqslant j \leqslant N$, define $\bar{\alpha}^{a,i,j} \in \mathbb{R}^{(1\times m)}$, $\bar{\beta}^{a,i,j} \in \mathbb{R}^{(m\times 1)}$ and $\bar{\mu}(a,i,j) \in \mathbb{R}$ to be the quantities computed by the conventional inside-outside algorithm in Figure 3 on the input $x_1 \dots x_N$. Define $\alpha^{a,i,j} \in \mathbb{R}^{(1\times m)}$, $\beta^{a,i,j} \in \mathbb{R}^{(m\times 1)}$ and $\mu(a,i,j) \in \mathbb{R}$ to be the quantities computed by the algorithm in Figure 3.*

Then for all $i \in [N]$, $\alpha^{a,i,j} = \bar{\alpha}^{a,i,j}(G^a)^{-1}$ and $\beta^{a,i,j} = G^a \bar{\beta}^{a,i,j}$. It follows that for all (a,i,j),

$$\mu(a,i,j) = \alpha^{a,i,j}\beta^{a,i,j} = \bar{\alpha}^{a,i,j}(G^a)^{-1}G^a\bar{\beta}^{a,i,j} = \bar{\alpha}^{a,i,j}\bar{\beta}^{a,i,j} = \bar{\mu}(a,i,j).$$

Hence the outputs from the algorithms in Figures 3 and 5 are the same, and it follows that the tensor-based algorithm in Figure 5 is correct.

Thus the vectors $\alpha^{a,i,j}$ and $\beta^{a,i,j}$ are linearly related to the vectors $\bar{\alpha}^{a,i,j}$ and $\bar{\beta}^{a,i,j}$, which are the inside and outside terms calculated by the conventional form of the inside-outside algorithm.

Proof: The proof is by induction, and is similar to the proof of Lemma 9.

First, we prove that the inside terms satisfy the relation $\alpha^{a,i,j} = \bar{\alpha}^{a,i,j}(G^a)^{-1}$.

The base case of the induction is as follows. By definition, for any $a \in \mathcal{P}, i \in [N], h \in [m]$, we have $\bar{\alpha}_h^{a,i,i} = q(a \to x_i | h, a)$. We also have for any $a \in \mathcal{P}, i \in [N]$, $\alpha^{a,i,i} = c^\infty_{a\to x_i} = q_{a\to x_i}(G^a)^{-1}$. It follows directly that $\alpha^{a,i,i} = \bar{\alpha}^{a,i,i}(G^a)^{-1}$ for any $a \in \mathcal{P}, i \in [N]$.

The inductive case is as follows. By definition, we have $\forall a \in \mathcal{I}, 1 \leqslant i < j \leqslant N, h \in [m]$

$$\bar{\alpha}_h^{a,i,j} = \sum_{k=i}^{j-1} \sum_{b,c} \sum_{h_2 \in [m]} \sum_{h_3 \in [m]} t(a \to b\ c, h_2, h_3 | h, a) \times \bar{\alpha}_{h_2}^{b,i,k} \times \bar{\alpha}_{h_3}^{c,k+1,j}.$$

We also have $\forall a \in \mathcal{I}, 1 \leqslant i < j \leqslant N$,

$$\begin{aligned}
\alpha^{a,i,j} &= \sum_{k=i}^{j-1} \sum_{b,c} C^{a\to b\ c}(\alpha^{b,i,k}, \alpha^{c,k+1,j}) && (23)\\
&= \sum_{k=i}^{j-1} \sum_{b,c} \left(T^{a\to b\ c}(\alpha^{b,i,k}G^b, \alpha^{c,k+1,j}G^c) \right)(G^a)^{-1} && (24)\\
&= \sum_{k=i}^{j-1} \sum_{b,c} \left(T^{a\to b\ c}(\bar{\alpha}^{b,i,k}, \bar{\alpha}^{c,k+1,j}) \right)(G^a)^{-1} && (25)\\
&= \bar{\alpha}^{a,i,j}(G^a)^{-1}. && (26)
\end{aligned}$$

Eq. 23 follows by the definitions in algorithm 5. Eq. 24 follows by the assumption in the theorem that

$$C^{a\to b\ c}(y^1, y^2) = \left(T^{a\to b\ c}(y^1 G^b, y^2 G^c) \right)(G^a)^{-1}$$

Eq. 25 follows because by the inductive hypothesis,

$$\alpha^{b,i,k} = \bar{\alpha}^{b,i,k}(G^b)^{-1}$$

and

$$\alpha^{c,k+1,j} = \bar{\alpha}^{c,k+1,j}(G^c)^{-1}.$$

Eq. 26 follows because

$$\left[T^{a\to b\ c}(\bar{\alpha}^{b,i,k}, \bar{\alpha}^{c,k+1,j})\right]_h = \sum_{h_2,h_3} t(a \to b\ c, h_2, h_3|h, a)\bar{\alpha}^{b,i,k}_{h_2}\bar{\alpha}^{c,k+1,j}_{h_3}$$

hence

$$\sum_{k=i}^{j-1}\sum_{b,c} T^{a\to b\ c}(\bar{\alpha}^{b,i,k}, \bar{\alpha}^{c,k+1,j}) = \bar{\alpha}^{a,i,j}.$$

We now turn the outside terms, proving that $\beta^{a,i,j} = G^a\bar{\beta}^{a,i,j}$. The proof is again by induction.

The base case is as follows. By the definitions in the algorithms, for all $a \in \mathcal{I}$, $\beta^{a,1,n} = c^1_a = G^a\pi^a$, and for all $a \in \mathcal{I}, h \in [m]$, $\bar{\beta}^{a,1,n}_h = \pi(a,h)$. It follows directly that for all $a \in \mathcal{I}$, $\beta^{a,1,n} = G^a\bar{\beta}^{a,1,n}$.

The inductive case is as follows. By the definitions in the algorithms, we have $\forall a \in \mathcal{N}, 1 \leqslant i \leqslant j \leqslant N, h \in [m]$

$$\bar{\beta}^{a,i,j}_h = \gamma^{1,a,i,j}_h + \gamma^{2,a,i,j}_h$$

where

$$\gamma^{1,a,i,j}_h = \sum_{k=1}^{i-1}\sum_{b\to c\ a}\sum_{h_2\in[m]}\sum_{h_3\in[m]} t(b \to c\ a, h_3, h|h_2, b) \times \bar{\beta}^{b,k,j}_{h_2} \times \bar{\alpha}^{c,k,i-1}_{h_3}$$

$$\gamma^{2,a,i,j}_h = \sum_{k=j+1}^{N}\sum_{b\to a\ c}\sum_{h_2\in[m]}\sum_{h_3\in[m]} t(b \to a\ c, h, h_3|h_2, b) \times \bar{\beta}^{b,i,k}_{h_2} \times \bar{\alpha}^{c,j+1,k}_{h_3}$$

and $\forall a \in \mathcal{N}, 1 \leqslant i \leqslant j \leqslant N$,

$$\beta^{a,i,j} = \sum_{k=1}^{i-1}\sum_{b\to c\ a} C^{b\to c\ a}_{(1,2)}(\beta^{b,k,j}, \alpha^{c,k,i-1}) + \sum_{k=j+1}^{N}\sum_{b\to a\ c} C^{b\to a\ c}_{(1,3)}(\beta^{b,i,k}, \alpha^{c,j+1,k}).$$

Critical identities are

$$\sum_{k=1}^{i-1}\sum_{b\to c\ a} C^{b\to c\ a}_{(1,2)}(\beta^{b,k,j}, \alpha^{c,k,i-1}) = G^a\gamma^{1,a,i,j} \quad (27)$$

$$\sum_{k=j+1}^{N}\sum_{b\to a\ c} C^{b\to a\ c}_{(1,3)}(\beta^{b,i,k}, \alpha^{c,j+1,k}) = G^a\gamma^{2,a,i,j} \quad (28)$$

from which $\beta^{a,i,j} = G^a\bar{\beta}^{a,i,j}$ follows immediately.

The identities in Eq. 27 and 28 are proved through straightforward algebraic manipulation, based on the following properties:

- By the inductive hypothesis, $\beta^{b,k,j} = G^b \bar{\beta}^{b,k,j}$ and $\beta^{b,i,k} = G^b \bar{\beta}^{b,i,k}$.
- By correctness of the inside terms, as shown earlier in this proof, it holds that $\alpha^{c,k,i-1} = \bar{\alpha}^{c,k,i-1}(G^c)^{-1}$ and $\alpha^{c,j+1,k} = \bar{\alpha}^{c,j+1,k}(G^c)^{-1}$.
- By the assumptions in the theorem,

$$C^{a \to b\ c}(y^1, y^2) = \left(T^{a \to b\ c}(y^1 G^b, y^2 G^c)\right)(G^a)^{-1}.$$

It follows (see Lemma 11) that

$$\begin{aligned} C^{b \to c\ a}_{(1,2)}(\beta^{b,k,j}, \alpha^{c,k,i-1}) &= G^a \left(T^{b \to c\ a}_{(1,2)}((G^b)^{-1}\beta^{b,k,j}, \alpha^{c,k,i-1} G^c)\right) \\ &= G^a \left(T^{b \to c\ a}_{(1,2)}(\bar{\beta}^{b,k,j}, \bar{\alpha}^{c,k,i-1})\right) \end{aligned}$$

and

$$C^{b \to a\ c}_{(1,3)}(\beta^{b,i,k}, \alpha^{c,j+1,k}) = G^a \left(T^{b \to a\ c}_{(1,3)}(\bar{\beta}^{b,i,k}, \bar{\alpha}^{c,j+1,k})\right)$$

■

Finally, we give the following Lemma, as used above:

Lemma 11 *Assume we have tensors $C \in \mathbb{R}^{m \times m \times m}$ and $T \in \mathbb{R}^{m \times m \times m}$ such that for any y^2, y^3,*

$$C(y^2, y^3) = \left(T(y^2 A, y^3 B)\right) D$$

where A, B, D are matrices in $\mathbb{R}^{m \times m}$. Then for any y^1, y^2,

$$C_{(1,2)}(y^1, y^2) = B\left(T_{(1,2)}(Dy^1, y^2 A)\right) \tag{29}$$

and for any y^1, y^3,

$$C_{(1,3)}(y^1, y^3) = A\left(T_{(1,3)}(Dy^1, y^3 B)\right). \tag{30}$$

Proof: Consider first Eq. 29. We will prove the following statement:

$$\forall y^1, y^2, y^3, \quad y^3 C_{(1,2)}(y^1, y^2) = y^3 B\left(T_{(1,2)}(Dy^1, y^2 A)\right)$$

This statement is equivalent to Eq. 29.

First, for all y^1, y^2, y^3, by the assumption that $C(y^2, y^3) = \left(T(y^2 A, y^3 B)\right) D$,

$$C(y^2, y^3) y^1 = T(y^2 A, y^3 B) D y^1$$

hence

$$\sum_{i,j,k} C_{i,j,k} y^1_i y^2_j y^3_k = \sum_{i,j,k} T_{i,j,k} z^1_i z^2_j z^3_k \tag{31}$$

where $z^1 = Dy^1$, $z^2 = y^2 A$, $z^3 = y^3 B$.

In addition, it is easily verified that

$$y^3 C_{(1,2)}(y^1, y^2) = \sum_{i,j,k} C_{i,j,k} y^1_i y^2_j y^3_k \quad (32)$$

$$y^3 B\left(T_{(1,2)}(Dy^1, y^2 A)\right) = \sum_{i,j,k} T_{i,j,k} z^1_i z^2_j z^3_k \quad (33)$$

where again $z^1 = Dy^1$, $z^2 = y^2 A$, $z^3 = y^3 B$. Combining Eqs. 31, 32, and 33 gives

$$y^3 C_{(1,2)}(y^1, y^2) = y^3 B\left(T_{(1,2)}(Dy^1, y^2 A)\right),$$

thus proving the identity in Eq. 29.

The proof of the identity in Eq. 30 is similar, and is omitted for brevity. ■

A.2 Proof of the Identity in Eq. 17

We now prove the identity in Eq. 17, repeated here:

$$D^{a \to b\ c}(y^1, y^2) = \left(T^{a \to b\ c}(y^1 G^b, y^2 G^c)\right) \mathrm{diag}(\gamma^a)(K^a)^\top.$$

Recall that

$$D^{a \to b\ c} = \mathbf{E}\left[[\![R_1 = a \to b\ c]\!] Z Y_2^\top Y_3^\top | A_1 = a\right],$$

or equivalently

$$D^{a \to b\ c}_{i,j,k} = \mathbf{E}\left[[\![R_1 = a \to b\ c]\!] Z_i Y_{2,j} Y_{3,k} | A_1 = a\right].$$

Using the chain rule, and marginalizing over hidden variables, we have

$$\begin{aligned} D^{a \to b\ c}_{i,j,k} &= \mathbf{E}\left[[\![R_1 = a \to b\ c]\!] Z_i Y_{2,j} Y_{3,k} | A_1 = a\right] \\ &= \sum_{h_1, h_2, h_3 \in [m]} p(a \to b\ c, h_1, h_2, h_3 | a) \mathbf{E}\left[Z_i Y_{2,j} Y_{3,k} | R_1 = a \to b\ c, h_1, h_2, h_3\right]. \end{aligned}$$

By definition, we have

$$p(a \to b\ c, h_1, h_2, h_3 | a) = \gamma^a_{h_1} \times t(a \to b\ c, h_2, h_3 | h_1, a)$$

In addition, under the independence assumptions in the L-PCFG, and using the definitions of K^a and G^a, we have

$$\begin{aligned} & \mathbf{E}\left[Z_i Y_{2,j} Y_{3,k} | R_1 = a \to b\ c, h_1, h_2, h_3\right] \\ = \ & \mathbf{E}\left[Z_i | A_1 = a, H_1 = h_1\right] \times \mathbf{E}\left[Y_{2,j} | A_2 = b, H_2 = h_2\right] \times \mathbf{E}\left[Y_{3,k} | A_3 = c, H_3 = h_3\right] \\ = \ & K^a_{i,h_1} \times G^b_{j,h_2} \times G^c_{k,h_3}. \end{aligned}$$

Putting this all together gives

$$\begin{aligned} D^{a \to b\ c}_{i,j,k} &= \sum_{h_1, h_2, h_3 \in [m]} \gamma^a_{h_1} \times t(a \to b\ c, h_2, h_3 | h_1, a) \times K^a_{i,h_1} \times G^b_{j,h_2} \times G^c_{k,h_3} \\ &= \sum_{h_1 \in [m]} \gamma^a_{h_1} \times K^a_{i,h_1} \times \sum_{h_2, h_3 \in [m]} t(a \to b\ c, h_2, h_3 | h_1, a) \times G^b_{j,h_2} \times G^c_{k,h_3}. \end{aligned}$$

By the definition of tensors,

$$\begin{aligned}
& [D^{a\to b\ c}(y^1, y^2)]_i \\
= & \sum_{j,k} D^{a\to b\ c}_{i,j,k} y^1_j y^2_k \\
= & \sum_{h_1\in[m]} \gamma^a_{h_1} \times K^a_{i,h_1} \times \sum_{h_2,h_3\in[m]} t(a \to b\ c, h_2, h_3|h_1, a) \times \left(\sum_j y^1_j G^b_{j,h_2}\right) \times \left(\sum_k y^2_k G^c_{k,h_3}\right) \\
= & \sum_{h_1\in[m]} \gamma^a_{h_1} \times K^a_{i,h_1} \times \left[T^{a\to b\ c}(y^1 G^b, y^2 G^c)\right]_{h_1}. \qquad (34)
\end{aligned}$$

The last line follows because by the definition of tensors,

$$\left[T^{a\to b\ c}(y^1 G^b, y^2 G^c)\right]_{h_1} = \sum_{h_2,h_3} T^{a\to b\ c}_{h_1,h_2,h_3} \left[y^1 G^b\right]_{h_2} \left[y^2 G^c\right]_{h_3}$$

and we have

$$\begin{aligned}
T^{a\to b\ c}_{h_1,h_2,h_3} &= t(a \to b\ c, h_2, h_3|h_1, a) \\
\left[y^1 G^b\right]_{h_2} &= \sum_j y^1_j G^b_{j,h_2} \\
\left[y^2 G^c\right]_{h_3} &= \sum_k y^2_k G^c_{k,h_3}.
\end{aligned}$$

Finally, the required identity

$$D^{a\to b\ c}(y^1, y^2) = \left(T^{a\to b\ c}(y^1 G^b, y^2 G^c)\right) \mathrm{diag}(\gamma^a)(K^a)^\top$$

follows immediately from Eq. 34. ■

A.3 Proof of the Identity in Eq. 18

We now prove the identity in Eq. 18, repeated below:

$$d^\infty_{a\to x} = q_{a\to x} \mathrm{diag}(\gamma^a)(K^a)^\top.$$

Recall that by definition

$$d^\infty_{a\to x} = \mathbf{E}\left[[\![R_1 = a \to x]\!] Z^\top | A_1 = a\right],$$

or equivalently

$$[d^\infty_{a\to x}]_i = \mathbf{E}\left[[\![R_1 = a \to x]\!] Z_i | A_1 = a\right].$$

Marginalizing over hidden variables, we have

$$\begin{aligned}
[d^\infty_{a\to x}]_i &= \mathbf{E}\left[[\![R_1 = a \to x]\!] Z_i | A_1 = a\right] \\
&= \sum_h p(a \to x, h|a) \mathbf{E}[Z_i | H_1 = h, R_1 = a \to x].
\end{aligned}$$

By definition, we have

$$p(a \to x, h|a) = \gamma_h^a q(a \to x|h, a) = \gamma_h^a \left[q_{a\to x}\right]_h .$$

In addition, by the independence assumptions in the L-PCFG, and the definition of K^a,

$$\mathbf{E}[Z_i|H_1 = h, R_1 = a \to x] = \mathbf{E}[Z_i|H_1 = h, A_1 = a] = K_{i,h}^a.$$

Putting this all together gives

$$\left[d_{a\to x}^\infty\right]_i = \sum_h \gamma_h^a \left[q_{a\to x}\right]_h K_{i,h}^a$$

from which the required identity

$$d_{a\to x}^\infty = q_{a\to x}\text{diag}(\gamma^a)(K^a)^\top$$

follows immediately. ■

A.4 Proof of the Identity in Eq. 19

We now prove the identity in Eq. 19, repeated below:

$$\Sigma^a = G^a\text{diag}(\gamma^a)(K^a)^\top$$

Recall that by definition

$$\Sigma^a = \mathbf{E}[Y_1 Z^\top|A_1 = a]$$

or equivalently

$$\left[\Sigma^a\right]_{i,j} = \mathbf{E}[Y_{1,i} Z_j|A_1 = a]$$

Marginalizing over hidden variables, we have

$$\begin{aligned}\left[\Sigma^a\right]_{i,j} &= \mathbf{E}[Y_{1,i} Z_j|A_1 = a] \\ &= \sum_h p(h|a)\mathbf{E}[Y_{1,i} Z_j|H_1 = h, A_1 = a]\end{aligned}$$

By definition, we have

$$\gamma_h^a = p(h|a)$$

In addition, under the independence assumptions in the L-PCFG, and using the definitions of K^a and G^a, we have

$$\begin{aligned}\mathbf{E}[Y_{1,i} Z_j|H_1 = h, A_1 = a] &= \mathbf{E}[Y_{1,i}|H_1 = h, A_1 = a] \times \mathbf{E}[Z_j|H_1 = h, A_1 = a] \\ &= G_{i,h}^a K_{j,h}^a\end{aligned}$$

Putting all this together gives

$$\left[\Sigma^a\right]_{i,j} = \sum_h \gamma_h^a G_{i,h}^a K_{j,h}^a$$

from which the required identity

$$\Sigma^a = G^a\text{diag}(\gamma^a)(K^a)^\top$$

follows immediately. ■

A.5 Proof of the Identity in Eq. 20

We now prove the identity in Eq. 19, repeated below:

$$c^1_a = G^a \pi^a.$$

Recall that by definition

$$c^1_a = \mathbf{E}\left[[\![A_1 = a]\!] Y_1 | B = 1\right],$$

or equivalently

$$[c^1_a]_i = \mathbf{E}\left[[\![A_1 = a]\!] Y_{1,i} | B = 1\right].$$

Marginalizing over hidden variables, we have

$$\begin{aligned}[c^1_a]_i &= \mathbf{E}\left[[\![A_1 = a]\!] Y_{1,i} | B = 1\right] \\ &= \sum_h P(A_1 = a, H_1 = h | B = 1) \mathbf{E}\left[Y_{1,i} | A_1 = a, H_1 = h, B = 1\right].\end{aligned}$$

By definition we have

$$P(A_1 = a, H_1 = h | B = 1) = \pi(a, h)$$

By the independence assumptions in the PCFG, and the definition of G^a, we have

$$\begin{aligned}\mathbf{E}\left[Y_{1,i} | A_1 = a, H_1 = h, B = 1\right] &= \mathbf{E}\left[Y_{1,i} | A_1 = a, H_1 = h\right] \\ &= G^a_{i,h}.\end{aligned}$$

Putting this together gives

$$[c^1_a]_i = \sum_h \pi(a, h) G^a_{i,h}$$

from which the required identity

$$c^1_a = G^a \pi^a$$

follows. ■

Appendix B. Proof of Theorem 8

In this section we give a proof of Theorem 8. The proof relies on three lemmas:

- In Section B.1 we give a lemma showing that if estimates $\hat{C}^{a \to b\ c}$, $\hat{c}_{a \to x}$ and $\hat{c}^1_a$ are close (up to linear transforms) to the parameters of an L-PCFG, then the distribution defined by the parameters is close (in l_1-norm) to the distribution under the L-PCFG.
- In Section B.2 we give a lemma showing that if the estimates $\hat{\Omega}^a$, $\hat{D}^{a \to b\ c}$, $\hat{d}^{\infty}_{a \to x}$ and $\hat{c}^1_a$ are close to the underlying values being estimated, the estimates $\hat{C}^{a \to b\ c}$, $\hat{c}_{a \to x}$ and $\hat{c}^1_a$ are close (up to linear transforms) to the parameters of the underlying L-PCFG.
- In Section B.3 we give a lemma relating the number of samples in the estimation algorithm to the errors in estimating $\hat{\Omega}^a$, $\hat{D}^{a \to b\ c}$, $\hat{d}^{\infty}_{a \to x}$ and $\hat{c}^1_a$.

The proof of the theorem is then given in Section B.4.

B.1 A Bound on How Errors Propagate

In this section we show that if estimated tensors and vectors $\hat{C}^{a \to b\ c}$, $\hat{c}^{\infty}_{a \to x}$ and $\hat{c}^{1}_{a}$ are sufficiently close to the underlying parameters $T^{a \to b\ c}$, $q^{\infty}_{a \to x}$, and π^a of an L-PCFG, then the distribution under the estimated parameters will be close to the distribution under the L-PCFG. Section B.1.1 gives assumptions and definitions; Lemma 12 then gives the main lemma; the remainder of the section gives proofs.

B.1.1 Assumptions and Definitions

We make the following assumptions:

- Assume we have an L-PCFG with parameters $T^{a \to b\ c} \in \mathbb{R}^{m \times m \times m}$, $q_{a \to x} \in \mathbb{R}^m$, $\pi^a \in \mathbb{R}^m$. Assume in addition that we have an invertible matrix $G^a \in \mathbb{R}^{m \times m}$ for each $a \in \mathcal{N}$. For convenience define $H^a = (G^a)^{-1}$ for all $a \in \mathcal{N}$.
- We assume that we have parameters $\hat{C}^{a \to b\ c} \in \mathbb{R}^{m \times m \times m}$, $\hat{c}^{\infty}_{a \to x} \in \mathbb{R}^{1 \times m}$ and $\hat{c}^{1}_{a} \in \mathbb{R}^{m \times 1}$ that satisfy the following conditions:
  - There exists some constant $\Delta > 0$ such that for all rules $a \to b\ c$, for all $y^1, y^2 \in \mathbb{R}^m$,
  $$||\hat{C}^{a \to b\ c}(y^1 H^b, y^2 H^c) G^a - T^{a \to b\ c}(y^1, y^2)||_\infty \leqslant \Delta ||y^1||_2 ||y^2||_2.$$
  - There exists some constant $\delta > 0$ such that for all $a \in \mathcal{P}$, for all $h \in [m]$,
  $$\sum_x |[\hat{c}^{\infty}_{a \to x} G^a]_h - [q^{\infty}_{a \to x}]_h| \leqslant \delta.$$
  - There exists some constant $\kappa > 0$ such that for all a,
  $$||(G^a)^{-1} \hat{c}^{1}_{a} - \pi^a||_1 \leqslant \kappa.$$

  We give the following definitions:

- For any skeletal tree $t = r_1 \dots r_N$, define $b^i(t)$ to be the quantities computed by the algorithm in Figure 4 with t together with the parameters $T^{a \to b\ c}$, $q^{\infty}_{a \to x}$, π^a as input. Define $\hat{f}^i(t)$ to be the quantities computed by the algorithm in Figure 4 with t together with the parameters $\hat{C}^{a \to b\ c}$, $\hat{c}^{\infty}_{a \to x}$, $\hat{c}^{1}_{a}$ as input. Define
  $$\xi(t) = b^1(t),$$
  and
  $$\hat{\xi}(t) = f^1(t) G^{a_1}.$$
  where as before a_1 is the non-terminal on the left-hand-side of rule r_1. Define $\hat{p}(t)$ to be the value returned by the algorithm in Figure 4 with t together with the parameters $\hat{C}^{a \to b\ c}$, $\hat{c}^{\infty}_{a \to x}$, $\hat{c}^{1}_{a}$ as input. Define $p(t)$ to be the value returned by the algorithm in Figure 4 with t together with the parameters $T^{a \to b\ c}$, $q^{\infty}_{a \to x}$, π^a as input.
- Define $\mathcal{T}(a, N)$ to be the set of of all skeletal trees with N binary rules (hence $2N + 1$ rules in total), with non-terminal a at the root of the tree.

- Define

$$\begin{aligned} Z(a,h,N) &= \sum_{t \in \mathcal{T}(a,N)} [\xi(t)]_h, \\ D(a,h,N) &= \sum_{t \in \mathcal{T}(a,N)} |[\hat{\xi}(t)]_h - [\xi(t)]_h|, \\ F(a,h,N) &= \frac{D(a,h,N)}{Z(a,h,N)}. \end{aligned}$$

- Define

$$\gamma = \min_{a,b,c \in \mathcal{N}, h_1,h_2,h_3 \in [m]} t(a \rightarrow b\ c, h_2, h_3 | a, h_1).$$

- For any $a \rightarrow b\ c$ define the tensor

$$\hat{T}^{a \rightarrow b\ c}(y^1, y^2) = \hat{C}^{a \rightarrow b\ c}(y^1 H^b, y^2 H^c) G^a.$$

B.1.2 The Main Lemma

Lemma 12 *Given the assumptions in Section B.1.1, for any a, N,*

$$\sum_{t \in \mathcal{T}(a,N)} |\hat{p}(t) - p(t)| \leqslant m \left((1+\kappa) \left(1 + \frac{\Delta}{\gamma} \right)^{N-1} (1+\delta)^N - 1 \right). \tag{35}$$

Proof: By definition we have

$$\begin{aligned} \sum_{t \in \mathcal{T}(a,N)} |\hat{p}(t) - p(t)| &= \sum_{t \in \mathcal{T}(a,N)} \left| \sum_h [\hat{\xi}(t)]_h [(G^a)^{-1} \hat{c}^1_a]_h - \sum_h [\xi(t)]_h \pi^a_h \right| \\ &= \sum_{t \in \mathcal{T}(a,N)} \left| \hat{\xi}(t) \cdot [(G^a)^{-1} \hat{c}^1_a] - \xi(t) \cdot \pi^a \right|. \end{aligned}$$

Define $e = [(G^a)^{-1} \hat{c}^1_a] - \pi^a$. Then by the triangle inequality,

$$\left| \hat{\xi}(t) \cdot [(G^a)^{-1} \hat{c}^1_a] - \xi(t) \cdot \pi^a \right| \leqslant |\hat{\xi}(t) \cdot \pi^a - \xi(t) \cdot \pi^a| + |\hat{\xi}(t) \cdot e - \xi(t) \cdot e| + |\xi(t) \cdot e|$$

We bound each of the three terms as follows:

$$|\hat{\xi}(t) \cdot \pi^a - \xi(t) \cdot \pi^a| \leqslant ||\hat{\xi}(t) - \xi(t)||_\infty ||\pi^a||_1 \leqslant ||\hat{\xi}(t) - \xi(t)||_\infty \leqslant \sum_h \left| [\hat{\xi}(t)]_h - [\xi(t)]_h \right|$$

$$|\hat{\xi}(t) \cdot e - \xi(t) \cdot e| \leqslant ||\hat{\xi}(t) - \xi(t)||_\infty ||e||_1 \leqslant \kappa ||\hat{\xi}(t) - \xi(t)||_\infty \leqslant \kappa \sum_h \left| [\hat{\xi}(t)]_h - [\xi(t)]_h \right|$$

$$|\xi(t) \cdot e| \leqslant ||\xi(t)||_\infty ||e||_1 \leqslant \kappa ||\xi(t)||_\infty \leqslant \kappa \sum_h [\xi(t)]_h.$$

Combining the above gives

$$\begin{aligned}\sum_{t\in\mathcal{T}(a,N)}|\hat{p}(t)-p(t)| &\leqslant (1+\kappa)\sum_{t\in\mathcal{T}(a,N)}\sum_h\left|[\hat{\xi}(t)]_h-[\xi(t)]_h\right|+\kappa\sum_{t\in\mathcal{T}(a,N)}\sum_h[\xi(t)]_h\\ &\leqslant m(1+\kappa)\left(\left(1+\frac{\Delta}{\gamma}\right)^N(1+\delta)^{N+1}-1\right)+m\kappa\\ &= m\left((1+\kappa)\left(1+\frac{\Delta}{\gamma}\right)^N(1+\delta)^{N+1}-1\right)\end{aligned}$$

where the second inequality follows because $\sum_{t\in\mathcal{T}(a,N)}\sum_h[\xi(t)]_h\leqslant m$, and because Lemma 13 gives

$$\sum_{t\in\mathcal{T}(a,N)}\sum_h\left|[\hat{\xi}(t)]_h-[\xi(t)]_h\right|\leqslant m\left(\left(1+\frac{\Delta}{\gamma}\right)^N(1+\delta)^{N+1}-1\right).$$

■

We now give a crucial lemma used in the previous proof:

Lemma 13 *Given the assumptions in Section B.1.1, for any a, h, N,*

$$D(a,h,N)=\sum_{t\in\mathcal{T}(a,N)}\left|[\hat{\xi}(t)]_h-[\xi(t)]_h\right|\leqslant Z(a,h,N)\left(\left(1+\frac{\Delta}{\gamma}\right)^N(1+\delta)^{N+1}-1\right).$$

Proof: A key identity is the following, which holds for any $N\geqslant 1$ (recall that $F(a,h,N)=D(a,h,N)/Z(a,h,N)$):

$$\begin{aligned}&F(a,h,N)\\ \leqslant\ &-1+\sum_{k=0}^{N-1}\sum_{b,c}\sum_{h_1,h_2}g(a,b,c,k,h_1,h_2)(1+F(b,h_1,k))(1+F(c,h_2,N-k-1))\\ &+\Delta\frac{Y(N)}{Z(a,h,N)}\sum_{k=0}^{N-1}\sum_{b,c}\sum_{h_1,h_2}h(b,c,k,h_1,h_2)(1+F(b,h_1,k))(1+F(c,h_2,N-k-1)),\end{aligned}\tag{36}$$

where

$$\begin{aligned}g(a,b,c,k,h_1,h_2) &= t(a\to b\ c,h_1,h_2|a,h)\frac{Z(b,h_1,k)Z(c,h_2,N-k-1)}{Z(a,h,N)}\\ Y(N) &= \sum_{k=0}^{N-1}\sum_{b,c}\sum_{h_1,h_2}Z(b,h_1,k)Z(c,h_2,N-k-1)\\ h(b,c,k,h_1,h_2) &= \frac{Z(b,h_1,k)Z(c,h_2,N-k-1)}{Y(N)}.\end{aligned}$$

The proof of Eq. 36 is in Section B.1.3. Note that we have

$$\sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} g(a,b,c,k,h_1,h_2) = \sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} h(b,c,k,h_1,h_2) = 1.$$

The rest of the proof follows through induction. For the base case, for $N = 0$ we have

$$Z(a,h,N)\left(\left(1+\frac{\Delta}{\gamma}\right)^N (1+\delta)^{N+1} - 1\right) = \delta Z(a,h,N) = \delta$$

where the last equality follows because $Z(a,h,0) = 1$ for any a, h. For $N = 0$ we also have

$$\sum_{t \in \mathcal{T}(a,N)} \left| [\hat{\xi}(t)]_h - [\xi(t)]_h \right| = \sum_x |[\hat{c}^{\infty}_{a \to x} G^a]_h - [q^{\infty}_{a \to x}]_h| \leqslant \delta.$$

The base case follows immediately.

For the recursive case, by the inductive hypothesis we have

$$1 + F(b,h_1,k) \leqslant \left(1+\frac{\Delta}{\gamma}\right)^k (1+\delta)^{k+1}$$

and

$$1 + F(c,h_2,N-k-1) \leqslant \left(1+\frac{\Delta}{\gamma}\right)^{N-k-1} (1+\delta)^{N-k}.$$

It follows from Eq. 36 that

$$\begin{aligned} F(a,h,N) &\leqslant -1 + \left(1 + \Delta \frac{Y(N)}{Z(a,h,N)}\right)\left(1+\frac{\Delta}{\gamma}\right)^{N-1} (1+\delta)^{N+1} \\ &\leqslant -1 + \left(1+\frac{\Delta}{\gamma}\right)^N (1+\delta)^{N+1} \end{aligned}$$

where the second inequality follows because

$$\frac{Y(N)}{Z(a,h,N)} = \frac{\sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} Z(b,h_1,k) Z(c,h_2,N-k-1)}{\sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} t(a \to b\ c, h_1, h_2 | a, h) Z(b,h_1,k) Z(c,h_2,N-k-1)} \leqslant \frac{1}{\gamma}.$$

This completes the proof. ■

B.1.3 Proof of Eq. 36

Any tree $t \in \mathcal{T}(a,N)$ where $N \geqslant 1$ can be decomposed into the following: 1) A choice b, c, implying the rule $a \to b\ c$ is at the root; 2) A choice of $0 \leqslant k \leqslant N-1$, implying that the tree dominated by b is of size k, the tree dominated by c is of size $N-1-k$; 3) A choice of trees $t_1 \in \mathcal{T}(b,k)$ and $t_2 \in \mathcal{T}(c,N-1-k)$. The resulting tree has $\xi_h(t) = T_h^{a \to b\ c}(\xi(t_1), \xi(t_2))$.

Define $d(t) = \xi(t) - \hat{\xi}(t)$. We then have the following:

$$\begin{aligned}
& \sum_{t \in \mathcal{T}(a,N)} |\hat{\xi}_h(t) - \xi_h(t)| \\
= & \sum_{k=0}^{N-1} \sum_{b,c} \sum_{t_1 \in \mathcal{T}(b,k)} \sum_{t_2 \in \mathcal{T}(c,N-1-k)} |\hat{T}_h^{a \to b\ c}(\hat{\xi}(t_1), \hat{\xi}(t_2)) - T_h^{a \to b\ c}(\xi(t_1), \xi(t_2))| \\
\leqslant & \Delta \sum_{k=0}^{N-1} \sum_{b,c} \sum_{t_1 \in \mathcal{T}(b,k)} \sum_{t_2 \in \mathcal{T}(c,N-1-k)} (||\xi(t_1)||_2 + ||d(t_1)||_2)(||\xi(t_2)||_2 + ||d(t_2)||_2) \\
& + \sum_{k=0}^{N-1} \sum_{b,c} \sum_{t_1 \in \mathcal{T}(b,k)} \sum_{t_2 \in \mathcal{T}(c,N-1-k)} |T_h^{a \to b\ c}(\xi(t_1), d(t_2))| \\
& + \sum_{k=0}^{N-1} \sum_{b,c} \sum_{t_1 \in \mathcal{T}(b,k)} \sum_{t_2 \in \mathcal{T}(c,N-1-k)} |T_h^{a \to b\ c}(d(t_1), \xi(t_2))| \\
& + \sum_{k=0}^{N-1} \sum_{b,c} \sum_{t_1 \in \mathcal{T}(b,k)} \sum_{t_2 \in \mathcal{T}(c,N-1-k)} |T_h^{a \to b\ c}(d(t_1), d(t_2))|. \qquad (37)
\end{aligned}$$

The inequality follows because by Lemma 14,

$$\begin{aligned}
& |\hat{T}_h^{a \to b\ c}(\hat{\xi}(t_1), \hat{\xi}(t_2)) - T_h^{a \to b\ c}(\xi(t_1), \xi(t_2))| \\
\leqslant & \Delta(||\xi(t_1)||_2 + ||d(t_1)||_2)(||\xi(t_2)||_2 + ||d(t_2)||_2) \\
& + |T_h^{a \to b\ c}(\xi(t_1), d(t_2))| + |T_h^{a \to b\ c}(d(t_1), \xi(t_2))| + |T_h^{a \to b\ c}(d(t_1), d(t_2))|.
\end{aligned}$$

We first derive an upper bound on the last three terms of Eq. 37. Note that we have the identity

$$\begin{aligned}
& Z(a, h, N) \\
= & \sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} t(a \to b\ c, h_1, h_2 | a, h) \sum_{t_1 \in \mathcal{T}(b,k)} \xi_{h_1}(t_1) \sum_{t_2 \in \mathcal{T}(c,N-1-k)} \xi_{h_2}(t_2) \\
= & \sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} t(a \to b\ c, h_1, h_2 | a, h) Z(b, h_1, k) Z(c, h_2, N-k-1).
\end{aligned}$$

It follows that

$$
\begin{aligned}
&\sum_{k=0}^{N-1}\sum_{b,c}\sum_{t_1\in\mathcal{T}(b,k)}\sum_{t_2\in\mathcal{T}(c,N-1-k)}\big(|T_h^{a\to b\ c}(\xi(t_1),d(t_2))|+|T_h^{a\to b\ c}(d(t_1),\xi(t_2))|\\
&\qquad\qquad +|T_h^{a\to b\ c}(d(t_1),d(t_2))|\big)\\
=\ &\sum_{k=0}^{N-1}\sum_{b,c}\sum_{h_1,h_2}t(a\to b\ c,h_1,h_2|a,h)\sum_{t_1\in\mathcal{T}(b,k)}\xi(t_1)_{h_1}\sum_{t_2\in\mathcal{T}(c,N-1-k)}|d(t_2)_{h_2}|\\
&+\sum_{k=0}^{N-1}\sum_{b,c}\sum_{h_1,h_2}t(a\to b\ c,h_1,h_2|a,h)\sum_{t_1\in\mathcal{T}(b,k)}|d(t_1)_{h_1}|\sum_{t_2\in\mathcal{T}(c,N-1-k)}\xi(t_2)_{h_2}\\
&+\sum_{k=0}^{N-1}\sum_{b,c}\sum_{h_1,h_2}t(a\to b\ c,h_1,h_2|a,h)\sum_{t_1\in\mathcal{T}(b,k)}|d(t_1)_{h_1}|\sum_{t_2\in\mathcal{T}(c,N-1-k)}|d(t_2)_{h_2}|\\
=\ &\Bigg(\sum_{k=0}^{N-1}\sum_{b,c}\sum_{h_1,h_2}t(a\to b\ c,h_1,h_2|a,h)\\
&\qquad\times\Bigg(\sum_{t_1\in\mathcal{T}(b,k)}\sum_{t_2\in\mathcal{T}(c,N-1-k)}(\xi(t_1)_{h_1}+|d(t_1)_{h_1}|)(\xi(t_2)_{h_2}+|d(t_2)_{h_2}|)\Bigg)\Bigg)-Z(a,h,N)\\
=\ &\Bigg(\sum_{k=0}^{N-1}\sum_{b,c}\sum_{h_1,h_2}t(a\to b\ c,h_1,h_2|a,h)(Z(b,h_1,k)+\\
&\qquad D(b,h_1,k))(Z(c,h_2,N-k-1)+D(c,h_2,N-k-1))\Bigg)-Z(a,h,N)\\
=\ &\Bigg(\sum_{k=0}^{N-1}\sum_{b,c}\sum_{h_1,h_2}t(a\to b\ c,h_1,h_2|a,h)Z(b,h_1,k)Z(c,h_2,N-k-1)\\
&\qquad\times(1+\frac{D(b,h_1,k)}{Z(b,h_1,k)})(1+\frac{D(c,h_2,N-k-1)}{Z(c,h_2,N-k-1)})\Bigg)-Z(a,h,N)\\
=\ &Z(a,h,N)\left(\sum_{k=0}^{N-1}\sum_{b,c}\sum_{h_1,h_2}g(a,b,c,k,h_1,h_2)(1+F(b,h_1,k))(1+F(c,h_2,N-k-1))\right)\\
&-Z(a,h,N)
\end{aligned}
\tag{38}
$$

where $g(a,b,c,k,h_1,h_2)=\frac{t(a\to b\ c,h_1,h_2|a,h)Z(b,h_1,k)Z(c,h_2,N-k-1)}{Z(a,h,n)}$.

We next derive a bound on the first term as follows:

$$\begin{aligned}
& \Delta \sum_{k=0}^{N-1} \sum_{b,c} \sum_{t_1 \in \mathcal{T}(b,k)} \sum_{t_2 \in \mathcal{T}(c,N-1-k)} (||\xi(t_1)||_2 + ||d(t_1)||_2)(||\xi(t_2)||_2 + ||d(t_2)||_2) \\
\leqslant & \Delta \sum_{k=0}^{N-1} \sum_{b,c} \sum_{t_1 \in \mathcal{T}(b,k)} \sum_{t_2 \in \mathcal{T}(c,N-1-k)} (||\xi(t_1)||_1 + ||d(t_1)||_1)(||\xi(t_2)||_1 + ||d(t_2)||_1) \\
= & \Delta \sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} (Z(b,h_1,k) + D(b,h_1,k))(Z(c,h_2,N-k-1) + D(c,h_2,N-k-1)) \\
= & \Delta \sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} Z(b,h_1,k)Z(c,h_2,N-k-1)(1+F(b,h_1,k))(1+F(c,h_2,N-k-1)) \\
= & \Delta Y(N) \sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} h(k,b,c,h_1,h_2)(1+F(b,h_1,k))(1+F(c,h_2,N-k-1)) \qquad (39)
\end{aligned}$$

where

$$h(k,b,c,h_1,h_2) = \frac{Z(b,h_1,k)Z(c,h_2,N-k-1)}{Y(N)}$$

and $Y(N) = \sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} Z(b,h_1,k)Z(c,h_2,N-k-1)$.

Combining Eqs. 37, 38 and 39 gives the inequality in Eq. 36, repeated below:

$$\begin{aligned}
& F(a,h,N) \\
\leqslant & -1 \\
& + \sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} g(a,b,c,k,h_1,h_2)(1+F(b,h_1,k))(1+F(c,h_2,N-k-1)) \\
& + \Delta \frac{Y(N)}{Z(a,h,N)} \sum_{k=0}^{N-1} \sum_{b,c} \sum_{h_1,h_2} h(b,c,k,h_1,h_2)(1+F(b,h_1,k))(1+F(c,h_2,N-k-1)).
\end{aligned}$$

■

The following lemma was used in the previous proof:

Lemma 14 *Assume we have tensors $\hat{T}$ and T and that there is some constant Δ such that for any $y^1, y^2 \in \mathbb{R}^m$,*

$$||\hat{T}(y^1,y^2) - T(y^1,y^2)||_\infty \leqslant \Delta ||y^1||_2 ||y^2||_2$$

Then for any $y^1, y^2, \hat{y}^1, \hat{y}^2$, for any h, it follows that

$$\begin{aligned}
|\hat{T}_h(\hat{y}^1,\hat{y}^2) - T_h(y^1,y^2)| \leqslant & \Delta(||y^1||_2 + ||d^1||_2)(||y^2||_2 + ||d^2||_2) \\
& + |T_h(y^1,d^2)| + |T_h(d^1,d^2)| + |T_h(d^1,y^2)|
\end{aligned}$$

where $d^1 = \hat{y}^1 - y^1$, and $d^2 = \hat{y}^2 - y^2$.

Proof: Define

$$\hat{g}(y^1) = \hat{T}_h(y^1, \hat{y}^2),$$
$$g(y^1) = T_h(y^1, y^2).$$

Define $d^1 = (\hat{y}^1 - y^1)$, $d^2 = (\hat{y}^2 - y^2)$. For any $v \in \mathbb{R}^m$,

$$\begin{aligned} |\hat{g}(v) - g(v)| &= |\hat{T}_h(v, \hat{y}^2) - T_h(v, y^2)| \\ &\leqslant |\hat{T}_h(v, y^2) - T_h(v, y^2)| + |\hat{T}_h(v, d^2) - T_h(v, d^2)| + |T_h(v, d^2)|. \end{aligned}$$

We can then derive the following bound:

$$\begin{aligned} &|\hat{T}_h(\hat{y}^1, \hat{y}^2) - T_h(y^1, y^2)| = |\hat{g}(\hat{y}^1) - g(y^1)| \\ &\leqslant |\hat{g}(y^1) - g(y^1)| + |\hat{g}(d^1) - g(d^1)| + |g(d^1)| \\ &\leqslant |\hat{T}_h(y^1, y^2) - T_h(y^1, y^2)| + |\hat{T}_h(y^1, d^2) - T_h(y^1, d^2)| + |T_h(y^1, d^2)| \\ &\quad + |\hat{T}_h(d^1, y^2) - T_h(d^1, y^2)| + |\hat{T}_h(d^1, d^2) - T_h(d^1, d^2)| + |T_h(d^1, d^2)| \\ &\quad + |T_h(d^1, y^2)| \\ &\leqslant \Delta(||y^1||_2 + ||d^1||_2)(||y^2||_2 + ||d^2||_2) \\ &\quad + |T_h(y^1, d^2)| + |T_h(d^1, d^2)| + |T_h(d^1, y^2)|. \end{aligned}$$

■

B.2 Relating Δ, δ, κ to Estimation Errors

We now give a lemma that relates estimation errors in the algorithm to the values for Δ, δ and κ as defined in the previous section.

Throughout this section, in addition to the estimates $\hat{D}^{a \to b\ c}$, $\hat{d}^{\infty}_{a \to x}$, $\hat{\Sigma}^a$, $\hat{C}^{a \to b\ c}$, $\hat{c}^{\infty}_{a \to x}$, $\hat{c}^1_a$ computed by the algorithm in Figure 7, we define quantities

$$\begin{aligned} \Sigma^a &= \mathbf{E}[Y_1 Z^\top | A_1 = a] \\ D^{a \to b\ c} &= \mathbf{E}\left[[\![R_1 = a \to b\ c]\!] Z Y_2^\top Y_3^\top | A_1 = a\right] \\ d^{\infty}_{a \to x} &= \mathbf{E}\left[[\![R_1 = a \to x]\!] Z^\top | A_1 = a\right] \\ C^{a \to b\ c}(y^1, y^2) &= D^{a \to b\ c}(y^1, y^2)(\Sigma^a)^{-1} \\ c^{\infty}_{a \to x} &= d^{\infty}_{a \to x}(\Sigma^a)^{-1} \\ c^1_a &= \mathbf{E}\left[[\![A_1 = a]\!] Y_1 | B = 1\right] \end{aligned}$$

where

$$Y_1 = (\hat{U}^{a_1})^\top \phi(T_1) \quad Z = (\hat{V}^{a_1})^\top \psi(O)$$
$$Y_2 = (\hat{U}^{a_2})^\top \phi(T_2) \quad Y_3 = (\hat{U}^{a_3})^\top \phi(T_3).$$

Note that these definitions are identical to those given in Section 7.2, with the additional detail that the projection matrices used to define random variables Y_1, Y_2, Y_3, Z are $\hat{U}^a$ and $\hat{V}^a$, that is, the projection matrices estimated in the first step of the algorithm in Figure 7.

The lemma is as follows:

Lemma 15 *Assume that under a run of the algorithm in Figure 7 there are constants* $\epsilon_\Omega^1, \epsilon_\Omega^2, \epsilon_D, \epsilon_d, \epsilon_\pi$ *such that*

$$\begin{aligned}
\forall a \in \mathcal{P}, \quad & ||\hat{\Omega}^a - \Omega^a||_F \leqslant \epsilon_\Omega^1 \\
\forall a \in \mathcal{I}, \quad & ||\hat{\Omega}^a - \Omega^a||_F \leqslant \epsilon_\Omega^2 \\
\forall a \to b\ c, \quad & ||\hat{D}^{a \to b\ c} - D^{a \to b\ c}||_F \leqslant \epsilon_D \\
\forall a \in \mathcal{P}, \quad & \sqrt{\sum_x ||\hat{d}^\infty_{a \to x} - d^\infty_{a \to x}||_2^2} \leqslant \epsilon_d \\
\forall a, \quad & ||\hat{c}_a^1 - c_a^1||_2 \leqslant \epsilon_\pi.
\end{aligned}$$

Assume in addition that $\epsilon_\Omega^1 \leqslant \min_{a \in \mathcal{P}} \frac{\sigma_m(\Omega^a)}{3}$ *and* $\epsilon_\Omega^2 \leqslant \min_{a \in \mathcal{I}} \frac{\sigma_m(\Omega^a)}{3}$. *For all* a *define* $G^a = (\hat{U}^a)^\top I^a$ *and* $H^a = (G^a)^{-1}$. *Then:*

- *For all* a, G^a *is invertible.*
- *For all* $y^1, y^2 \in \mathbb{R}^m$, *for all rules of the form* $a \to b\ c$

$$||\hat{C}^{a \to b\ c}(y^1 H^b, y^2 H^c) G^a - C^{a \to b\ c}(y^1 H^b, y^2 H^c) G^a||_\infty \leqslant \Delta ||y^1||_2 ||y^2||_2$$

  where

$$\Delta = \frac{16}{3} \frac{1}{\sigma_m(I^b)\sigma_m(I^c)} \left(\frac{\epsilon_\Omega^2}{\sigma_m(\Omega^a)^2} + \frac{\epsilon_D}{3\sigma_m(\Omega^a)} \right).$$

- *For all* $a \in \mathcal{P}$, *for all* $h \in [m]$,

$$\sum_x |[\hat{c}^\infty_{a \to x} G^a]_h - [c^\infty_{a \to x} G^a]_h| \leqslant \delta$$

  where

$$\delta = 4 \left(\frac{\epsilon_\Omega^1}{\sigma_m(\Omega^a)^2} + \frac{\epsilon_d \sqrt{n}}{3\sigma_m(\Omega^a)} \right).$$

- *For all* a,

$$||(G^a)^{-1} \hat{c}_a^1 - (G^a)^{-1} c_a^1||_1 \leqslant \kappa$$

  where

$$\kappa = \frac{2}{\sqrt{3}} \frac{\sqrt{m}}{\sigma_m(\Omega^a)} \epsilon_\pi.$$

B.2.1 Proof of Lemma 15

We first prove three necessary lemmas, then give a proof of Lemma 15.

Lemma 16 *Assume we have vectors and matrices* $d \in \mathbb{R}^{1 \times m}$, $\Sigma \in \mathbb{R}^{m \times m}$, $\hat{d} \in \mathbb{R}^{1 \times m}$, $\hat{\Sigma} \in \mathbb{R}^{m \times m}$, $U \in \mathbb{R}^{d \times m}$, $I \in \mathbb{R}^{d \times m}$. *We assume that* Σ, $\hat{\Sigma}$, *and* $(U^\top I)$ *are invertible.*

In addition, define

$$\begin{aligned}
c &= d\Sigma^{-1} \\
\hat{c} &= \hat{d}\hat{\Sigma}^{-1} \\
G^a &= U^\top I.
\end{aligned}$$

We assume:

- *For $h = 1 \ldots m$, $||I_h||_2 \leqslant 1$, where I_h is the h'th column of I^a.*
- *$||U||_{2,o} \leqslant 1$ where $||U||_{2,o}$ is the spectral norm of the matrix U.*
- $||\hat{\Sigma} - \Sigma||_{2,o} \leqslant \epsilon_1$

It follows that

$$||\hat{c}G^a - cG^a||_\infty \leqslant \frac{1+\sqrt{5}}{2} \frac{\epsilon_1 ||\hat{d}||_2}{\min\{\sigma_m(\Sigma), \sigma_m(\hat{\Sigma})\}^2} + \frac{||\hat{d} - d||_2}{\sigma_m(\Sigma)}.$$

Proof:

$$\begin{aligned}
& ||\hat{c}G^a - cG^a||_\infty \\
= \; & ||(\hat{c} - c)U^\top I||_\infty \\
& \text{(By definition } G^a = U^\top I) \\
\leqslant \; & ||(\hat{c} - c)U^\top||_2 \\
& \text{(By } ||I_h||_2 \leqslant 1) \\
\leqslant \; & ||\hat{c} - c||_2 \\
& \text{(By } ||U||_{2,o} \leqslant 1) \\
= \; & ||\hat{d}\hat{\Sigma}^{-1} - d\Sigma^{-1}||_2 \\
& \text{(By definitions of } c, \hat{c}) \\
\leqslant \; & ||\hat{d}(\hat{\Sigma}^{-1} - \Sigma^{-1})||_2 + ||(\hat{d} - d)\Sigma^{-1}||_2 \\
& \text{(By triangle inequality)} \\
\leqslant \; & ||\hat{d}||_2 ||\hat{\Sigma}^{-1} - \Sigma^{-1}||_{2,o} + ||\hat{d} - d||_2 ||\Sigma^{-1}||_{2,o} \\
& \text{(By definition of } ||.||_{2,o}) \\
\leqslant \; & ||\hat{d}||_2 \frac{1+\sqrt{5}}{2} \frac{\epsilon_1}{\min\{\sigma_m(\Sigma), \sigma_m(\hat{\Sigma})\}^2} + \frac{||\hat{d} - d||_2}{\sigma_m(\Sigma)} \\
& \text{(By Lemma 23 of Hsu et al. 2009, and } ||\Sigma^{-1}||_{2,o} = 1/\sigma_m(\Sigma))
\end{aligned}$$

Lemma 17 *Assume we have vectors $c, \hat{c} \in \mathbb{R}^{m \times 1}$, and we have a matrix $G^a \in \mathbb{R}^{m \times m}$ that is invertible. It follows that*

$$||(G^a)^{-1}\hat{c} - (G^a)^{-1}c||_1 \leqslant \frac{\sqrt{m}||\hat{c} - c||_2}{\sigma_m(G^a)}.$$

Proof:

$$||(G^a)^{-1}\hat{c} - (G^a)^{-1}c||_1 \leqslant \sqrt{m}||(G^a)^{-1}\hat{c} - (G^a)^{-1}c||_2 \leqslant \frac{\sqrt{m}||\hat{c} - c||_2}{\sigma_m(G^a)}.$$

The first inequality follows because $||.||_1 \leqslant \sqrt{m}||.||_2$. The second inequality follows because $||(G^a)^{-1}||_{2,o} = 1/\sigma_m(G^a)$.

Lemma 18 *Assume we have matrices and tensors $D \in \mathbb{R}^{m \times m \times m}$, $\Sigma \in \mathbb{R}^{m \times m}$, $\hat{D} \in \mathbb{R}^{m \times m \times m}$, $\hat{\Sigma} \in \mathbb{R}^{m \times m}$, $U \in \mathbb{R}^{d \times m}$, $I \in \mathbb{R}^{d \times m}$, $G^b \in \mathbb{R}^{m \times m}$, $G^c \in \mathbb{R}^{m \times m}$. We assume that Σ, $\hat{\Sigma}$, G^b, G^c, and $U^\top I$ are invertible.*

In addition define

$$\begin{aligned} C(y^1, y^2) &= D(y^1, y^2)\Sigma^{-1} \\ \hat{C}(y^1, y^2) &= \hat{D}(y^1, y^2)\hat{\Sigma}^{-1} \\ G^a &= U^\top I \\ H^b &= (G^b)^{-1} \\ H^c &= (G^c)^{-1} \end{aligned}$$

We assume:

- *For $h = 1 \ldots m$, $||I_h||_2 \leqslant 1$, where I_h is the h'th column of I^a.*
- $||U||_{2,o} \leqslant 1$
- $||\hat{\Sigma} - \Sigma||_{2,o} \leqslant \epsilon_1$

It follows that for any $y^1, y^2 \in \mathbb{R}^m$,

$$\begin{aligned} & ||\hat{C}(y^1 H^b, y^2 H^c)G^a - C(y^1 H^b, y^2 H^c)G^a||_\infty \\ \leqslant \; & \frac{||y^1||_2 ||y^2||_2}{\sigma_m(G^b)\sigma_m(G^c)} \left(\frac{1+\sqrt{5}}{2} \times \frac{\epsilon_1 ||\hat{D}||_F}{\min\{\sigma_m(\Sigma), \sigma_m(\hat{\Sigma})\}^2} + \frac{||\hat{D} - D||_F}{\sigma_m(\Sigma)} \right). \end{aligned}$$

Proof:

$$\begin{aligned} & ||\hat{C}(y^1 H^b, y^2 H^c)G^a - C(y^1 H^b, y^2 H^c)G^a||_\infty \\ \leqslant \; & ||\hat{D}(y^1 H^b, y^2 H^c)||_2 \frac{1+\sqrt{5}}{2} \frac{\epsilon_1}{\min\{\sigma_m(\Sigma), \sigma_m(\hat{\Sigma})\}^2} + \frac{||\hat{D}(y^1 H^b, y^2 H^c) - D(y^1 H^b, y^2 H^c)||_2}{\sigma_m(\Sigma)} \\ & \text{(By Lemma 16, using } \hat{d} = \hat{D}(y^1 H^b, y^2 H^c),\ d = D(y^1 H^b, y^2 H^c).) \\ \leqslant \; & ||y^1 H^b||_2 ||y^2 H^c||_2 \left(||\hat{D}||_F \frac{1+\sqrt{5}}{2} \frac{\epsilon_1}{\min\{\sigma_m(\Sigma), \sigma_m(\hat{\Sigma})\}^2} + \frac{||\hat{D} - D||_F}{\sigma_m(\Sigma)} \right) \\ & \text{(By } ||D(v^1, v^2)||_2 \leqslant ||D||_F ||v^1||_2 ||v^2||_2 \text{ for any tensor } D, \text{ vectors } v^1, v^2.) \\ \leqslant \; & \frac{||y^1||_2 ||y^2||_2}{\sigma_m(G^b)\sigma_m(G^c)} \left(||\hat{D}||_F \frac{1+\sqrt{5}}{2} \frac{\epsilon_1}{\min\{\sigma_m(\Sigma), \sigma_m(\hat{\Sigma})\}^2} + \frac{||\hat{D} - D||_F}{\sigma_m(\Sigma)} \right) \\ & \text{(By } H^b = (G^b)^{-1} \text{ hence } ||H^b||_{2,o} = 1/\sigma_m(G^b). \text{ Similar for } H^c.) \end{aligned}$$

Proof of Lemma 15: By Lemma 9 of Hsu et al. (2009), assuming that $\epsilon_\Omega \leqslant \min_a \frac{\sigma_m(\Omega^a)}{3}$ gives for all a

$$\sigma_m(\hat{\Sigma}^a) \geqslant \frac{2}{3}\sigma_m(\Omega^a)$$

$$\sigma_m(\Sigma^a) \geqslant \frac{\sqrt{3}}{2}\sigma_m(\Omega^a)$$

$$\sigma_m(G^a) \geqslant \frac{\sqrt{3}}{2}\sigma_m(I^a)$$

The condition that $\sigma_m(I^a) > 0$ implies that $\sigma_m(G^a) > 0$ and hence G^a is invertible. The values for Δ and κ follow from lemmas 18 and and 17 respectively.

The value for δ is derived as follows. By Lemma 16 we have for any rule $a \to x$, for any $h \in [m]$,

$$|[\hat{c}^{\infty}_{a\to x}G^a]_h - [c^{\infty}_{a\to x}G^a]_h| \leqslant \frac{1+\sqrt{5}}{2}\frac{\epsilon_1||\hat{d}^{\infty}_{a\to x}||_2}{\min\{\sigma_m(\Sigma^a), \sigma_m(\hat{\Sigma^a})\}^2} + \frac{||\hat{d}^{\infty}_{a\to x} - d^{\infty}_{a\to x}||_2}{\sigma_m(\Sigma^a)}. \quad (40)$$

By definition

$$\hat{d}^{\infty}_{a\to x} = \left(\frac{\sum_{i=1}^{M}[\![r^{(i,1)} = a \to x]\!]}{\sum_{i=1}^{M}[\![a_i = a]\!]}\right) \times \left(\frac{\sum_{i=1}^{M}[\![r^{(i,1)} = a \to x]\!](z^{(i)})^\top}{\sum_{i=1}^{M}[\![r^{(i,1)} = a \to x]\!]}\right)$$

In addition $z^{(i)} = (\hat{V}^{a_i})^\top\psi(t^{(i,1)})$ and $||\hat{V}^{a_i}||_{2,o} \leqslant 1$, $||\psi(t^{(i,1)})||_2 \leqslant 1$, hence $||z^{(i)}||_2 \leqslant 1$, and

$$||\hat{d}^{\infty}_{a\to x}||_2 \leqslant \frac{\sum_{i=1}^{M}[\![r^{(i,1)} = a \to x]\!]}{\sum_{i=1}^{M}[\![a_i = a]\!]}.$$

It follows that

$$\sum_x ||\hat{d}^{\infty}_{a\to x}||_2 \leqslant 1. \quad (41)$$

In addition we have

$$\sum_x ||\hat{d}^{\infty}_{a\to x} - d^{\infty}_{a\to x}||_2 \leqslant \sqrt{n}\sqrt{\sum_x ||\hat{d}^{\infty}_{a\to x} - d^{\infty}_{a\to x}||_2^2} \leqslant \sqrt{n}\epsilon_d. \quad (42)$$

Combining Eqs. 41, 42 and 40 gives for any $a \in \mathcal{P}$, for any $h \in [m]$,

$$\sum_x |[\hat{c}^{\infty}_{a\to x}G^a]_h - [c^{\infty}_{a\to x}G^a]_h| \leqslant \frac{1+\sqrt{5}}{2}\frac{\epsilon_1}{\min\{\sigma_m(\Sigma^a), \sigma_m(\hat{\Sigma^a})\}^2} + \frac{\sqrt{n}\sqrt{\sum_x ||\hat{d}^{\infty}_{a\to x} - d^{\infty}_{a\to x}||_2^2}}{\sigma_m(\Sigma^a)}$$

from which the lemma follows.

■

B.3 Estimation Errors

The next lemma relates estimation errors to the number of samples in the algorithm in Figure 4:

Lemma 19 *Consider the algorithm in Figure 7. With probability at least $1-\delta$, the following statements hold:*

$$\forall a \in \mathcal{I}, \sqrt{\sum_{b,c} ||\hat{D}^{a\to b\ c} - D^{a\to b\ c}||_F^2} \leqslant \sqrt{\frac{1}{M_a}} + \sqrt{\frac{2}{M_a}\log\frac{2|\mathcal{N}|+1}{\delta}}$$

$$\forall a \in \mathcal{P}, \sqrt{\sum_x ||\hat{d}^{\infty}_{a \to x} - d^{\infty}_{a \to x}||_2^2} \leqslant \sqrt{\frac{1}{M_a}} + \sqrt{\frac{2}{M_a} \log \frac{2|\mathcal{N}| + 1}{\delta}}$$

$$\forall a \in \mathcal{N}, ||\hat{\Omega}^a - \Omega^a||_F \leqslant \sqrt{\frac{1}{N_a}} + \sqrt{\frac{2}{N_a} \log \frac{2|\mathcal{N}| + 1}{\delta}}$$

$$\sqrt{\sum_a ||\hat{c}^1_a - c^1_a||_2^2} \leqslant \sqrt{\frac{1}{R}} + \sqrt{\frac{2}{R} \log \frac{2|\mathcal{N}| + 1}{\delta}}$$

B.3.1 Proof of Lemma 19

We first need the following lemma:

Lemma 20 *Assume i.i.d. random vectors $X_1 \dots X_N$ where each $X_i \in \mathbb{R}^d$, and for all i with probability 1, $||X_i||_2 \leqslant 1$. Define*

$$q = \mathbf{E}[X_i]$$

for all i and

$$\hat{Q} = \frac{\sum_{i=1}^N X_i}{N}.$$

Then for any $\epsilon > 0$,

$$\mathbf{P}(||\hat{Q} - q||_2 \geqslant 1/\sqrt{N} + \epsilon) \leqslant e^{-N\epsilon^2/2}.$$

Proof: The proof is very similar to the proof of proposition 19 of Hsu et al. (2009). Consider two random samples $x_1 \dots x_n$ and $y_1 \dots y_n$ where $x_i = y_i$ for all $i \neq k$. define

$$\hat{q} = \frac{\sum_{i=1}^N x_i}{N}$$

and

$$\hat{p} = \frac{\sum_{i=1}^N y_i}{N}.$$

Then

$$||\hat{q} - q||_2 - ||\hat{p} - q||_2 \leqslant ||\hat{q} - \hat{p}||_2 = \frac{||x_k - y_k||_2}{N} \leqslant \frac{||x_k||_2 + ||y_k||_2}{N} \leqslant \frac{2}{N}.$$

It follows through McDiarmid's inequality (McDiarmid, 1989) that

$$Pr(||\hat{Q} - q||_2 \geqslant \mathbf{E}||\hat{Q} - q||_2 + \epsilon) \leqslant e^{-N\epsilon^2/2}$$

In addition,

$$\begin{aligned}
& \mathbf{E}\left[||\hat{Q} - q||_2\right] \\
= \;& \mathbf{E}\left[||\frac{\sum_{i=1}^N X_i}{N} - q||_2\right] \\
= \;& \frac{1}{N}\mathbf{E}\left[||\sum_{i=1}^N (X_i - q)||_2\right] \\
\leqslant \;& \frac{1}{N}\sqrt{\mathbf{E}\left[||\sum_{i=1}^N (X_i - q)||_2^2\right]} \\
& \text{(By Jensen's inequality)} \\
= \;& \frac{1}{N}\sqrt{\sum_{i=1}^N \mathbf{E}\left[||(X_i - q)||_2^2\right]} \\
& \text{(By independence of the } X_i\text{'s)} \\
= \;& \frac{1}{N}\sqrt{\sum_{i=1}^N \mathbf{E}\left[||X_i||_2^2\right] - N||q||_2^2} \\
\leqslant \;& \frac{1}{N}\sqrt{N(1 - ||q||_2^2)} \\
& \text{(By } ||X_i||_2 \leqslant 1.\text{)} \\
\leqslant \;& \frac{1}{\sqrt{N}},
\end{aligned}$$

which completes the proof. ■

Proof of Lemma 19: For each $a \to b\ c$, $i, j, k \in [m]$, define a random variable

$$A_{i,j,k}^{a \to b\ c} = [\![R_1 = a \to b\ c]\!] Z_i Y_j^2 Y_k^3.$$

It follows that

$$D_{i,j,k}^{a \to b\ c} = \mathbf{E}[A_{i,j,k}^{a \to b\ c} | A_1 = a].$$

Note that

$$||Z||_2 = ||(V^a)^\top \psi(O)||_2 \leqslant 1$$

because $||V^a||_{2,o} \leqslant 1$, and $||\psi(O)||_2 \leqslant 1$. Similarly $||Y^2||_2 \leqslant 1$ and $||Y^3||_2 \leqslant 1$.

In addition we have for all $a \in \mathcal{I}$,

$$\begin{aligned}
\sum_{b,c}\sum_{i=1}^m\sum_{j=1}^m\sum_{k=1}^m |A_{i,j,k}^{a \to b\ c}|^2 &= \sum_{b,c}\sum_{i=1}^m\sum_{j=1}^m\sum_{k=1}^m |Z_i|^2 |Y_j^2|^2 |Y_k^3|^2 [\![R_1 = a \to b\ c]\!]^2 \\
&= ||Z||_2^2 ||Y^2||_2^2 ||Y^3||_2^2 (\sum_{b,c} [\![R_1 = a \to b\ c]\!]^2) \leqslant 1
\end{aligned}$$

It follows by an application of Lemma 20 that for the definitions of $D^{a\to b\ c}$ and $\hat{D}^{a\to b\ c}$ in Figure 7, for all a,

$$\mathbf{P}(\sqrt{\sum_{b,c}\sum_{i,j,k}|\hat{D}^{a\to b\ c}_{i,j,k}-D^{a\to b\ c}_{i,j,k}|^2}\geqslant 1/\sqrt{M_a}+\epsilon_1)\leqslant e^{-M_a\epsilon_1^2/2},$$

or equivalently,

$$\mathbf{P}\left(\sqrt{\sum_{b,c}||\hat{D}^{a\to b\ c}-D^{a\to b\ c}||_F^2}\geqslant\frac{1}{\sqrt{M_a}}+\sqrt{\frac{2}{M_a}\log\frac{2|\mathcal{N}|+1}{\delta}}\right)\leqslant\frac{\delta}{2|\mathcal{N}|+1}. \tag{43}$$

By a similar argument, if for each $a\in\mathcal{P}$, $x\in[n]$, $i\in[m]$ we define the random variable

$$B_i^{a\to x}=Z_i[\![R_1=a\to x]\!]$$

then

$$d^{\infty}_{a\to x}=\mathbf{E}[B_i^{a\to x}|A_1=a]$$

and

$$\sum_x\sum_{i=1}^m|B_i^{a\to x}|^2=\sum_x\sum_{i=1}^m|Z_i|^2[\![R_1=a\to x]\!]^2\leqslant 1$$

It follows by an application of Lemma 20 that for the definitions of $d^{\infty}_{a\to x}$ and $\hat{d}^{\infty}_{a\to x}$ in Figure 7, for all a,

$$\mathbf{P}(\sqrt{\sum_x\sum_i|[\hat{d}^{\infty}_{a\to x}]_i-[d^{\infty}_{a\to x}]_i|^2}\geqslant 1/\sqrt{N_a}+\epsilon_2)\leqslant e^{-M_a\epsilon_2^2/2}$$

or equivalently

$$\mathbf{P}\left(\sqrt{\sum_x||\hat{d}^{\infty}_{a\to x}-d^{\infty}_{a\to x}||_2^2}\geqslant\frac{1}{\sqrt{M_a}}+\sqrt{\frac{2}{M_a}\log\frac{2|\mathcal{N}|+1}{\delta}}\right)\leqslant\frac{\delta}{2|\mathcal{N}|+1}. \tag{44}$$

A similar argument can be used to show that for all a, for the definitions of Ω^a and $\hat{\Omega}^a$ in Figure 7,

$$\mathbf{P}(\sqrt{\sum_{i,j}|\hat{\Omega}^a_{i,j}-\Omega^a_{i,j}|^2}\geqslant 1/\sqrt{N_a}+\epsilon_3)\leqslant e^{-N_a\epsilon_3^2/2}$$

or equivalently

$$\mathbf{P}\left(||\hat{\Omega}^a-\Omega^a||_F\geqslant\frac{1}{\sqrt{N_a}}+\sqrt{\frac{2}{N_a}\log\frac{2|\mathcal{N}|+1}{\delta}}\right)\leqslant\frac{\delta}{2|\mathcal{N}|+1} \tag{45}$$

Finally, if we define the random variable

$$F_i^a=Y_i^1[\![A_1=a]\!]$$

then

$$\sum_a \sum_i |F_i^a|^2 = \sum_a \sum_i |Y_i^1|^2 [\![A_1 = a]\!]^2 \leqslant 1.$$

In addition

$$c_a^1 = \mathbf{E}[F_i^a | B = 1].$$

It follows by an application of Lemma 20 that for the definitions of c_a^1 and $\hat{c}_a^1$ in Figure 7,

$$\mathbf{P}(\sqrt{\sum_a \sum_i |[\hat{c}_a^1]_i - [c_a^1]_i|^2} \geqslant 1/\sqrt{R} + \epsilon_4) \leqslant e^{-R\epsilon_4^2/2}$$

or equivalently

$$\mathbf{P}\left(\sqrt{\sum_a ||\hat{c}_a^1 - c_a^1||_2^2} \geqslant \frac{1}{\sqrt{R}} + \sqrt{\frac{2}{R} \log \frac{2|\mathcal{N}|+1}{\delta}}\right) \leqslant \frac{\delta}{2|\mathcal{N}|+1}. \tag{46}$$

Finally, applying the union bound to the $2|\mathcal{N}|+1$ events in Eqs. 43, 44, 45 and 46 proves the theorem. ■

B.4 Proof of Theorem 8

Under the assumptions of the theorem, we have constants C_1, C_2, C_3, C_4 and C_5 such that

$$\forall a \in \mathcal{I}, N_a \geqslant L \times \left(C_1 \frac{N}{\gamma\epsilon} \frac{m}{\xi^2\sigma^2}\right)^2 \quad \forall a \in \mathcal{P}, N_a \geqslant L \times \left(\frac{C_2 N m}{\epsilon\sigma^2}\right)^2$$

$$\forall a \in \mathcal{I}, M_a \geqslant L \times \left(C_3 \frac{N}{\gamma\epsilon} \frac{m}{\xi^2\sigma}\right)^2 \quad \forall a \in \mathcal{P}, M_a \geqslant L \times \left(C_4 \frac{Nm\sqrt{n}}{\epsilon\sigma}\right)^2$$

$$R \geqslant L \times \left(C_5 \frac{Nm\sqrt{m}}{\epsilon\sigma}\right)^2$$

It follows from Lemma 19 that with probability at least $1-\delta$,

$$\begin{aligned}
\forall a \in \mathcal{I}, &\quad ||\hat{\Omega}^a - \Omega^a||_F \leqslant \epsilon_\Omega^1 \\
\forall a \in \mathcal{P}, &\quad ||\hat{\Omega}^a - \Omega^a||_F \leqslant \epsilon_\Omega^2 \\
\forall a \to b\ c, &\quad ||\hat{D}^{a \to b\ c} - D^{a \to b\ c}||_F \leqslant \epsilon_D \\
\forall a \in \mathcal{P}, &\quad \sqrt{\sum_x ||\hat{d}_{a \to x}^{\infty} - d_{a \to x}^{\infty}||_2} \leqslant \epsilon_d \\
\forall a, &\quad ||\hat{c}_a^1 - c_a^1||_2 \leqslant \epsilon_\pi
\end{aligned}$$

where

$$\epsilon_\Omega^1 \leqslant 3 \times \frac{1}{C_2} \times \sigma^2 \times \frac{\epsilon}{Nm}$$

$$\epsilon_\Omega^2 \leqslant 3 \times \frac{1}{C_1} \times \xi^2\sigma^2 \times \frac{\gamma\epsilon}{Nm}$$

$$\epsilon_D \leqslant 3 \times \frac{1}{C_3} \times \xi^2\sigma \times \frac{\gamma\epsilon}{Nm}$$

$$\epsilon_d \leqslant 3 \times \frac{1}{C_4} \times \sigma \times \frac{\epsilon}{\sqrt{n}Nm}$$
$$\epsilon_\pi \leqslant 3 \times \frac{1}{C_5} \times \frac{\sigma}{\sqrt{m}} \times \frac{\epsilon}{Nm}.$$

It follows from Lemma 15 that with suitable choices of $C_1 \dots C_5$, the inequalities in Lemma 15 hold with values

$$\Delta \leqslant \frac{\gamma\epsilon}{4Nm}$$
$$\delta \leqslant \frac{\epsilon}{4Nm}$$
$$\kappa \leqslant \frac{\epsilon}{4Nm}.$$

It follows from Lemma 12 that

$$\sum_{t \in \mathcal{T}(a,N)} |\hat{p}(t) - p(t)| \leqslant m \left(\left(1 + \frac{\epsilon}{4Nm} \right)^{2N} - 1 \right) \leqslant \epsilon$$

where the second inequality follows because $(1 + a/t)^t \leqslant 1 + 2a$ for $a \leqslant 1/2$.

References

S. Arora, R. Ge, Y. Halpern, D. M. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of ICML*, 2013.

R. Bailly, A. Habrar, and F. Denis. A spectral approach for probabilistic grammatical inference on trees. In *Proceedings of ALT*, 2010.

R. Bailly, Carreras P. X., F. M. Luque, and A. J. Quattoni. Unsupervised spectral learning of WCFG as low-rank matrix completion. In *Proceedings of EMNLP*, 2013.

J. Baker. Trainable grammars for speech recognition. In *Proceedings of ASA*, 1979.

B. Balle, A. Quattoni, and X. Carreras. A spectral learning algorithm for finite state transducers. In *Proceedings of ECML*, 2011.

P.J. Bickel and K.A. Doksum. *Mathematical Statistics: Basic Ideas And Selected Topics.* Mathematical Statistics: Basic Ideas and Selected Topics. Pearson Prentice Hall, 2006.

E. Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of AAAI-IAAI*, 1997.

S. B. Cohen and M. Collins. A provably correct learning algorithm for latent-variable PCFGs. In *Proceedings of ACL*, 2014.

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar. Spectral learning of latent-variable PCFGs. In *Proceedings of ACL*, 2012.

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar. Experiments with spectral learning of latent-variable PCFGs. In *Proceedings of NAACL*, 2013.

M. Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of ACL*, 1997.

S. Dasgupta. Learning mixtures of Gaussians. In *Proceedings of FOCS*, 1999.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

P. Dhillon, D. Foster, and L. Ungar. Multi-view learning of word embeddings via CCA. In *Proceedings of NIPS*, 2011.

P. Dhillon, J. Rodu, M. Collins, D. P. Foster, and L. H. Ungar. Spectral dependency parsing with latent variables. In *Proceedings of EMNLP*, 2012.

D. P. Foster, J. Rodu, and L. H. Ungar. Spectral dimensionality reduction for HMMs. arXiv:1203.6130, 2012.

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

D. Hsu, S. M. Kakade, and T. Zhang. A spectral algorithm for learning hidden Markov models. In *Proceedings of COLT*, 2009.

H. Jaeger. Observable operator models for discrete stochastic time series. *Neural Computation*, 12(6), 2000.

M. Johnson. PCFG models of linguistic tree representations.

D. Klein and C.D. Manning. Accurate Unlexicalized Parsing. In *Proceedings of ACL*, pages 423–430, 2003.

F. M. Luque, A. Quattoni, B. Balle, and X. Carreras. Spectral learning for non-deterministic dependency parsing. In *Proceedings of EACL*, 2012.

M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

T. Matsuzaki, Y. Miyao, and J. Tsujii. Probabilistic CFG with latent annotations. In *Proceedings of ACL*, 2005.

C. McDiarmid. On the method of bounded differences. *Surveys in Combinatorics*, pages 148–188, 1989.

A. Moitra and G. Valiant. Settling the polynomial learnability of mixtures of Gaussians. *IEEE Annual Symposium on Foundations of Computer Science*, pages 93–102, 2010. ISSN 0272-5428.

A. Parikh, L. Song, and E. P. Xing. A spectral algorithm for latent tree graphical models. In *Proceedings of ICML*, 2011.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of COLING-ACL*, 2006.

S. Siddiqi, B. Boots, and G. Gordon. Reduced-rank hidden Markov models. *Journal of Machine Learning Research*, 9:741–748, 2010.

L. Song, B. Boots, S. M. Siddiqi, G. J. Gordon, and A. J. Smola. Hilbert space embeddings of hidden Markov models. In *Proceedings of ICML*, 2010.

L. Song, A. P. Parikh, and E. P. Xing. Kernel embeddings of latent tree graphical models. In *NIPS*, pages 2708–2716, 2011.

K. Stratos, A. M. Rush, S. B. Cohen, and M. Collins. Spectral learning of refinement HMMs. In *Proceedings of CoNLL*, 2013.

S. A. Terwijn. On the learnability of hidden Markov models. In *Grammatical Inference: Algorithms and Applications (Amsterdam, 2002)*, volume 2484 of *Lecture Notes in Artificial Intelligence*, pages 261–268, Berlin, 2002. Springer.

A. Tropp, N. Halko, and P. G. Martinsson. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. In *Technical Report No. 2009-05*, 2009.

L. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

S. Vempala and G. Wang. A spectral algorithm for learning mixtures of distributions. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.