A Survey of Spectral Learning Approaches for Structured Prediction

Kartik Goyal
kartikgo@cs.cmu.edu

November 26, 2013

Abstract

Latent variable models(LVM) are prevalent for dealing with problems pertaining to Structured Prediciton as they offer flexibility in modelling observed data by introducing conditional independence between observed variables. Generally, algorithms for such models employ local search heuristics like EM algorithm, which have shortcomings like slow convergence, susceptibility to get stuck in local minima etc. Spectral methods offer a new perspective, which is significantly different from the optimization perspective, to view a LVM and provide fast and local minima free algorithms to estimate the objective. A lot of work has been done on development of spectral algorithms for latent variable models where the objective is to calculate marginal or conditional probabilities of observed variables, and estimation of model parameters is not necessary. However, some spectral algorithms have been developed for parameter estimation too. Embedding probability distributions in Hilbert spaces to model observations drawn from continuous distributions lead to nonparametric kernel based estimation algorithms. This survey focusses on spectral algorithms that have been developed for various models prevalent in structured prediction problems like Hidden Markov Models, Latent PCFG trees, Weighted Automata etc. Bulk of this survey covers the models where the observations are drawn from discrete distributions, however, an algorithm for HMMs having contiuous observed data will also be discussed.

1 Introduction

1.1 Latent Variable Models

Latent variable models are probabilistic models that relate a set of observed variables to an additional set of unobserved or hidden variables. These hidden variables can introduce conditional independence between the observed variables given the latent variables, which allows us to compute the marginal distribution of the observed variables by integrating out the latent variables. Hence, we can represent complex distributions of interrelated observed variables more tractably by modelling them with latent variables. Also, introducing latent variables offers flexibility of probabilistic modeling and helps in addressing a diverse range of problem such as topic modeling(Blei et al., 2003), social network modeling(Hoff et al., 2002) etc.

1.2 Expectation Maximization

Exact inference of model parameters which maximizes the likelihood of LVM is generally intractable because the number of possible configurations tends to grow exponentially with the size of input for mnay simple and useful LVMs. Therefore, learning the model parameters has relied on likelihood maximization and local search heuristics such as expectation maximization(EM)(Dempster et al., 1977).
EM is an iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing

data. Each iteration of the EM algorithm consists of two processes: The E-step, and the M-step. In the expectation, or E-step, the missing data are estimated given the observed data and current estimate of the model parameters. In the M-step, the likelihood function is maximized using the expectation of the missing data. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration. If the observed data is denoted by $\mathbf{X}$, hidden data by $\mathbf{z}$ and parameters by θ, EM maximizes the difference of likelihoods obtained by parameters at each iteration and it has the form:

$$\theta_{n+1} = argmax_{\theta}\, \mathbb{E}_{\mathbf{z}|\mathbf{X},\theta}\, ln(P(\mathbf{X}, \mathbf{z}|\theta))$$

Unfortunately, EM has a few drawbacks, as it is slow in convergence and requires many iterations. It slows down dramatically with the increase in dimensionality of data. Also, it can get stuck in bad local minima and may require a lot of reinitializations for a reliable estimate of parameters.

1.3 Spectral Methods

In many LVM problems, we are not interested in obtaining all the parameters and latent variables in the model and are only interested in computing the marginal probability of observed data or prediction of observable data. For these problems, we can view the problem from a perspective different than the optimization perspective of EM. The spectral perspective makes use of the connection between the LVMs and low rank factorization of matrices. The spectral view offers fast, local minima-free and consistent estimates however, it does not aim to find MLE.

1.3.1 Intuition

Consider two observable variables $\mathbf{A}$ and $\mathbf{B}$ which can take on m values each. The rank of the matrix for the joint probability $\mathcal{P}(\mathbf{A}, \mathbf{B})$ depends upon the dependence between $\mathbf{A}$ and $\mathbf{B}$. If both are independent, the rank will be 1. On the other hand, complete dependence will cause the matrix to have rank m. Instead of this clique we can assume a hidden variable $\mathbf{H}$ having k($\leq m$) states to be the parent of both $\mathbf{A}$ and $\mathbf{B}$. Now, the joint probability matrix can be factorized and the rank of this matrix will be $\leq k$.
This intuition of latent variables enabling low rank matrix factorizations drives the spectral algorithms.

1.3.2 Observable Representation

In the example above the joint probability matrix $\mathcal{P}((A, B))$ factorizes into three matrices: $\mathcal{P}(\mathbf{A}|\mathbf{H})$,$diag(\mathcal{P}(H))$ and $\mathcal{P}(\mathbf{B}|\mathbf{H})^T$.
The above factorization is not of great help as it involves hidden variables which we want to avoid estimating. The key is to convert the factorization into probability matrices of observables.[1]
In the following example, figure 1 is an HMM and joint probability $\mathcal{P}(\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}, \mathbf{X_4})$ is to be computed.

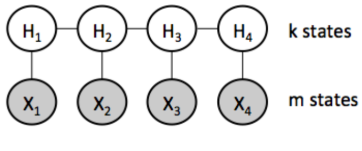


Figure 1: HMM

$$\mathbf{M} = \mathcal{P}(\mathbf{X_{1,2}}, \mathbf{X_{3,4}}) = \mathcal{P}(\mathbf{X_{1,2}}|\mathbf{H_2})diag(\mathcal{P}(\mathbf{H_2}))\mathcal{P}(\mathbf{X_{3,4}}|\mathbf{H_2})^T$$

[1]The following example is motivated by the slides at `http://www.cs.cmu.edu/~epxing/Class/10708/lecture/lecture21-spectral.pdf`

This factorization can be written as $\mathbf{M} = \mathbf{RL}$ where $\mathbf{R}$ is an $\mathrm{m}^2\times\mathrm{k}$ matrix and $\mathbf{L}$ is a $\mathrm{k}\times\mathrm{m}^2$.
But, this factorization is not unique. For any invertible matrix $\mathbf{S}$, $\mathbf{M} = \mathbf{RSS^{-1}L}$ is also a valid factorization. Hence, the key here is to find an alternative factorization that depends only on observed variables.
In the above example,
$\mathbf{R} = \mathcal{P}(\mathbf{X_{1,2}}, \mathbf{X_{3,4}}) = \mathcal{P}(\mathbf{X_{1,2}}|\mathbf{H_2})diag(\mathcal{P}(\mathbf{H_2}))$ and,
$\mathbf{L} = \mathcal{P}(\mathbf{X_{3,4}}|\mathbf{H_2})^T$
If we choose $\mathbf{S}$ to be $\mathcal{P}(\mathbf{X_3}|\mathbf{H_2})$,

$$\mathbf{M} = \mathbf{RSS^{-1}L} = \mathcal{P}(\mathbf{X_{1,2}}, \mathbf{X_{3,4}}) = \mathcal{P}(\mathbf{X_{1,2}}, \mathbf{X_3})\mathcal{P}(\mathbf{X_2}, \mathbf{X_3})^{-1}\mathcal{P}(\mathbf{X_2}, \mathbf{X_{3,4}}) \tag{1}$$

We can see that the above form is an observable factorization of the marginal probability and this idea is salient for most of the spectral algorithms discussed in this survey. Interestingly, for this method to work, the inverse must exist, which enforces a condition to be satisfied for this method to work.

$$\mathcal{P}(\mathbf{X_2}, \mathbf{X_3}) = \mathcal{P}(\mathbf{X_2}|\mathbf{H_2})diag(\mathcal{P}(H_2))\mathcal{P}(\mathbf{X_3}|\mathbf{H_2})^T$$

All matrices on the RHS must have full rank for the inverse to exist. If $\mathrm{m}\geq\mathrm{k}$, finding inverse is not hard if we project observable to lower dimensions. But if $m > k$, the inverse will not be compatible with the choice of $\mathbf{S}$ and generally the problem becomes intractable if $k \gg m$

2 General Framework for Spectral Algorithms

2.1 Notation

A tensor is a multi-dimensional array and it order is the number of dimensions, also called modes. Matrices and vectors are 2-mode and 1-mode tensors respectively. The representation in this survey is borrowed from (Song et al., 2011). Tensors of order 1 are represetnted by boldfaced lowercase letters e.g. $\mathbf{a}$. Tensors of order 2 are represented by boldfaced capital letters e.g. $\mathbf{A}$. Higher order tensors are represented by caligraphic letters e.g. $\mathcal{T}$ amd the scalars are represented by lower case letters.
A 'fiber' or slice of a tensor is obtained by fixing every index but one. Hence, the mode-n fiber of N-order tensor $\mathcal{T}$ is denoted as $\mathcal{T}(i_1, i_2, \ldots, i_{n-1}, :, i_{n+1}, \ldots, i_N)$.

Multiplication of Tensors

We use standard notation for order-0,1,2 tensors e.g. $\mathbf{AB}$, $\mathbf{Ab}$ etc.
In all the algorithms discussed, the only interesting multiplications of higher order tensors are with matrices and vectors. Let $\mathcal{T} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ be and Nth order tensor and $\mathbf{A} \in \mathbb{R}^{J\times I_n}$ be a matrix. Then,

$$\mathcal{T}' = \mathcal{T} \times_n \mathbf{A} \in \mathbb{R}^{I_1\times\ldots I_{n-1}\times J\times I_{n+1}\cdots\times I_N}$$

where entries $\mathcal{T}'(i_1, \ldots i_{n-1}, j, i_{n+1}, \ldots i_N)$ are defined as $\sum_{i_n=1}^{I_n} \mathcal{T}(i_1, \ldots i_{n-1}, i_n, i_{n+1}, \ldots i_N)\mathbf{A}(j, i_n)$
Let $\mathbf{a} \in \mathbb{R}^{I_n}$. Then,

$$\mathcal{T}' = \mathcal{T} \times_n \mathbf{a} \in \mathbb{R}^{I_1\times\ldots I_{n-1}\times I_{n+1}\cdots\times I_N}$$

where entries $\mathcal{T}'(i_1, \ldots i_{n-1}, i_{n+1}, \ldots i_N)$ are defined as $\sum_{i_n=1}^{I_n} \mathcal{T}(i_1, \ldots i_{n-1}, i_n, i_{n+1}, \ldots i_N)\mathbf{a}(i_n)$.
Hence, n-mode vector product reduces the dimensions of the tensor by 1 and n-mode matrix product does not change the order of the tensor.
Also,

$$\mathcal{T} \times_n \mathbf{A} \times_m \mathbf{B} = \mathcal{T} \times_m \mathbf{B} \times_n \mathbf{A}$$

and,

$$\mathcal{T} \times_n \mathbf{A} \times_n \mathbf{B} = \mathcal{T} \times_n (\mathbf{BA})$$

2.2 General Algorithm

The spectral algorithms work with the assumption of natural seperation condition with respect to the hidden states which requires that the latent states for a hidden variable are independent of each other. This implies that all the parameters and artifacts of the models are of rank m where m is the number of latent states(**Condition 1**). It can be relaxed for some of the parameters as long as the final factorization is of rank m.
With this condition, a typical spectral algorithm involves following steps:

1. The first step is to identify the set of parameters(dependent on latent states) of the models. Let us call this set θ.
2. The next step is to write the quanitity in which we are insterested(marginal or conditional observable probability) in terms of the original parameters, θ.
3. Develop an algorithm such that matrices obtained after Singular Value Decomposition(SVD) of some directly observable quantity, can be used to obtain surrogate parameters($\theta^{'}$) for the model which are observable representations and do not depend on any latent states.
4. Use these surrogate parameters to obtain the desired quantities.

As pointed out earlier, the crux of the algorithm lies in step 3, where observable representations are obtained in the form of surrogate parameters using matrix factorization. The algorithm must ensure that the matrix introduced to get an alternate factorization is invertible(**Condition 2**).
These algorithms gives consistent estimates if both the conditions are met. It should be noted that surrogate parameters and matrix factors are of rank m.

PAC learning

Most of the algorithms discussed in this survey include the sample complexity analysis. The proofs of these analyses are outside the scope of this survey. In general, the sample complexity depends upon the smallest singular value of the matrix which is factorized on the algorithm.

Evaluation

The standard evaluation of these algorithms is based upon comparison with EM algorithm with respect to speed and performance(error).

3 Models

This section describes various models used for Structured Prediction for which Spectral Algorithms have been developed till date. Specific models like Hidden Markov Models(HMM) and Latent PCGFs(L-PCFG) are discussed first. Later, spectral algotihtms for more general models like Weighted Finite Automata(WFA) and Latent Tree Models are described.

3.1 Hidden Markov Models

(Hsu et al., 2012) introduced a spectral algorithm for HMMs which aims to find either the joint probability of the observed sequence or the conditional distribution of a future observation, conditioned on some history of observations. It consists of hidden states(h_t) and observation states(x_t). The set of hidden states is denoted by $[m] = \{1, \ldots, m\}$ and observed states is denoted as $[n] = \{1, \ldots, n\}$ where $m \leq n$.

It assumes the natural seperation condition to be satisfied which implies that Transition matrix($\mathbf{T} \in \mathbb{R}^{m \times m}$) and Observation matrix($\mathbf{O} \in \mathbb{R}^{n \times m}$) and initial state distribution $\tilde{\pi} \in \mathbb{R}^m$ are rank m.(**Condition 1**).

The original parameters of HMM are $\mathbf{T}, \mathbf{O}, \tilde{\pi}$. They define $\mathbf{A}_x \forall x \in [n]$:

$$\mathbf{A}_x = \mathbf{T} diag(\mathbf{O}(x, :))$$

such that for any t:

$$Pr[x_1, \ldots, x_t] = \mathbf{\tilde{1}_m^T A_{x_t}} \ldots \mathbf{A_{x_1}} \tilde{\pi}$$

Observables

In their theoretical model, they analyze an algorithm that use intial few observations of the sequence and ignore the rest. They are marginal probabilities of observation singletons,pairs and triples which are defined $\forall x \in [n]$:

- $[P_1]_i = \mathbf{Pr}[x_1 = i]$
- $[P_{2,1}]_{ij} = \mathbf{Pr}[x_2 = i, x_1 = j]$
- $[P_{3,x,1}]_{ij} = \mathbf{Pr}[x_3 = i, x_2 = x, x_1 = j]$

where $\mathbf{P}_1 \mathbb{R}^n, \mathbf{P}_{2,1} \in \mathbb{R}^{n \times n} and P_{3,x,1} \in \mathbb{R}^{n \times n}$

Observable Representation

Further a matrix $\mathbf{U} \in \mathbb{R}^{n \times m}$ is introduced such that it defines an m-dimensional subspace that preserves the state dynamics. It is the matrix of left singular vectors of $P_{2,1}$ corresponding to non-zero singular values. It is proved that $\mathbf{U^T O}$ is invertible.(**Condition 2**)
Based upon these quantities surrogate parameters are defined as follows:

- $\mathbf{\tilde{b_1}} = \mathbf{U^T P_1} = (\mathbf{U^T O} \vec{\pi})$
- $\mathbf{\tilde{b}_\infty} = (\mathbf{P_{2,1}^T U})^+ \mathbf{P_1} = \mathbf{\tilde{1}_m} (\mathbf{U^T}))^{-1}$
- $\mathbf{B_x} = (\mathbf{U^T P_{3,x,1}})(\mathbf{U^T P_{2,1}})^+ \ \forall x \in [n]$
- $\mathbf{\tilde{b}_{t+1}} = \frac{\mathbf{B_{x_t} \tilde{b_t}}}{\mathbf{\tilde{b_\infty}^T B_{x_t} \tilde{b_t}}}$

These surrogate parameters result in an alternate factorization which is only dependent on observables. Using empirical estimate of $\mathbf{P}$ matrices and surrogate parameters based on the empirical quantities, following quantities can be estimated:

- $\mathbf{Pr}[x_1, \ldots, x_t] = \mathbf{b_\infty B_{x_t}} \ldots \mathbf{B_{x_1} b_1}$
- $\mathbf{Pr}[x_t | x_{1:t-1}] = \frac{\mathbf{b_\infty^T B_{x_t} \tilde{b_t}}}{\sum_x \mathbf{\tilde{b_\infty}^T B_{x_t} \tilde{b_t}}}$

The running time of the above algorithm is dominated by the SVD computation of an $n \times n$ matrix. Inference time is similar to normal HMM i.e. $\mathcal{O}(tm^2)$.

3.2 Latent-Variable PCFG

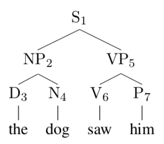
In this section, the spectral algorithm developed by Cohen et al. (2013) for learning of latent-variable PCFGs(Matsuzaki et al., 2005).

L-PCFG : Defniniton

L-PCFG is a PCFG where latent variables are assciated with the non-terminals. These latent variables enable the featurization of the standard inside-outside algorithm for parsing PCFGs. An L-PCFG is a 5-tuple (N,I,P,m,n) where:

- N is the set of non-terminals in the Grammar. $I \in N$ is a finite set of in-terminals. $P \in N$ is a set of preterminals. $N = P \cup I$ and $I \cap P = \emptyset$.
- $[m]$ is the set of hidden states.
- $[n]$ is the set of words.
- $\forall a \in I, b \in N, c \in N, h1, h2, h3 \in [m]$, there is a context free rule $a(h_1) \rightarrow b(h_2)c(h_3)$
- $\forall a \in P, h \in [m], x \in [n]$, there is a context free rule $a(h) \rightarrow x$.

A skeletal tree(s-tree) refers to a sequence of rules $r_1 \dots r_N$, which don't have any latent variables associated with the non-terminals. A full tree is a s-tree with latent values associated with the non-terminals.



r_1 = S → NP VP
r_2 = NP → D N
r_3 = D → the
r_4 = N → dog
r_5 = VP → V P
r_6 = V → saw
r_7 = P → him

Figure 2: A s-tree and its sequence of rules

Original Parameters

- For each $a \rightarrow bc$, there exists a diagonal matrix $\mathbf{Q}^{a\rightarrow bc} \in \mathbb{R}^{m\times m}$ with values $q(a \rightarrow bc|h, a)$.
- For each $a \rightarrow x$, there exists a diagonal matrix $\mathbf{Q}^{a\rightarrow x} \in \mathbb{R}^{m\times m}$ with values $q(a \rightarrow x|h, a)$(score of (a,h) at LHS of a rule)
- For each $a \rightarrow bc$, there exists a matrix $\mathbf{S}^{a\rightarrow bc} \in \mathbb{R}^{m\times m}$ where $\mathbf{S}_{h',h} = s(h'|h, a \rightarrow bc)$(score of left child and latent variable)
- For each $a \rightarrow bc$, there exists a matrix $\mathbf{T}^{a\rightarrow bc} \in \mathbb{R}^{m\times m}$ where $\mathbf{T}_{h',h} = t(h'|h, a \rightarrow bc)$(score of right child and latent variable)
- For each $a \in I$, there exists a vector $\pi^{\mathbf{a}} \in \mathbb{R}^m$ filled with the scores (h,a) at the root of the tree.

3.2.1 Objectives

Their spectral algorithm focuses on computing:

- $p(r_1, \dots, r_N)$ for a given s-tree.
- Given an input sequence $x = x_1 \dots x_N$, the marginal probabilities:
  $\mu(a, i, j) = \sum_{\tau\in\mathcal{T}:(a,i,j) in \tau} p(\tau)$
  for each non-terminal $a \in N, foreach(i, j)$ such that $1 \leq i \leq j \leq N$

3.2.2 Alternate Parametrization

The following parameters can be used for estimating the objective quantities.

- A Tensor $\mathcal{C}^{a\to bc} \in \mathbb{R}^{m\times m\times m}$ for each rule $a \to bc$
- A vector $\mathbf{c}^{\infty}_{a\to x} \in \mathbb{R}^{1\times m}$ for each rule $a \to x$
- A vector $\mathbf{c}^{1}_{a} \in \mathbb{R}^{m\times 1}$ for each $a \in I$

The first quantity can be computed by a recursive algorithm which just keeps on multiplying score in a bottom-up manner.The second quantity can be derived directly from the inside-outside algorithm which depends just on the above parameters. These algorithm are [presented in detail by (Cohen et al., 2013). These parameters also serve as the surrgoate parameters in the spectral algorithms. However, to derive the observable representations of these, various intermediate entities are introduced.

Observable representations

The observed data are the s-trees and hence the random variables for each rule r_i are R(rules),T_1(inside tree head node),T_2(inside tree of left child),T_3(inside tree of right child),H_1, H_2, H_3(hidden variables associate with T_1, T_2, T_3),O(outside tree),A_1, A_2, A_3(labels),B(indicator of root).
ψ maps outside trees *o* to feature vectors $\psi(o) \in \mathbb{R}^{d'}$ such that $d' \geq m$.
ϕ maps inside trees *t* to feature vectors $\phi(t) \in \mathbb{R}^{d}$ such that $d \geq m$.
Now these matrices $\forall a \in N$ are defined:
$I^a \in \mathbb{R}^{d\times m}$ such that $I^a_{i,h} = \mathbb{E}\left[\phi_i(T_1)|H_1 = h, A_1 = a\right]$
$J^a \in \mathbb{R}^{d'\times m}$ such that $J^a_{i,h} = \mathbb{E}\left[\psi_i(O)|H_1 = h, A_1 = a\right]$
$\gamma^a \in \mathbb{R}^m$ to denote $P(H|A_1)$
The seperability condition now states that $\forall a \in N$, the matrices $\mathbf{I}^a$ and $\mathbf{J}^a$ are of full rank m and $\gamma^a_h > 0$ (**Condition 1**)
If condition 1 holds then, projection matrices $\mathbf{U}^a \in \mathbb{R}^{d\times m}$ and $\mathbf{V}^a \in \mathbb{R}^{d'\times m} \forall a \in N$ are defined as follows: $\forall a \in N$, define $\Omega^a = \mathbb{E}\left[\phi(T_1)\psi(O)^T|A_1 = a\right]$. Then $\mathbf{U}^a$ is a matrix of m left singular vectors of Ω^a and $\mathbf{V}^a$ is a matrix of m right singular vectors of Ω^a
Now, it can be proven that $\mathbf{G}^a = (\mathbf{U}^a)^T\mathbf{I}^a$ and $\mathbf{K}^a = (\mathbf{V}^a)^T\mathbf{J}^a$ are invertible.(**Condition 2**)
Having obtained the projection matrices, following variables are defined:

- $\mathbf{Y}_1 = (\mathbf{U}^{a_1})^T\phi(T_1)$
- $\mathbf{Y}_2 = (\mathbf{U}^{a_2})^T\phi(T_2)$
- $\mathbf{Y}_3 = (\mathbf{U}^{a_3})^T\phi(T_3)$
- $\mathbf{Z} = (\mathbf{V}^{a_1})^T\phi(O)$

Following entities are now computed using the above vectors:

- $\Sigma^a = \mathbb{E}[\mathbf{Y}_1\mathbf{Z}^T|A_1 = a] \qquad \forall a \in N$
- $\mathcal{D}^{a\to bc} = \mathbb{E}[\mathbf{Y_3Z^TY_2^T}|A_1 = a] \qquad \forall$ rules $a \to bc$
- $\mathbf{d^{\infty}_{a\to x}} = \mathbb{E}[\mathbf{Z}^T|A_1 = a] \qquad \forall$ rules $a \to x$

Now it can be proven that if Condition 1 and Condition 2 are satisfied, then the alternate parameters can be computed as:

- $\mathcal{C}^{a \to bc} = \mathcal{D}^{a \to bc}(\Sigma^a)^{-1}$
- $\mathbf{c}^{\infty}_{a \to x} = \mathbf{d}^{\infty}_{a \to x}(\Sigma^a)^{-1}$
- $\mathbf{c}^1_a = \mathbb{E}[Y_1 | B = 1] \; \forall A_1 = a$

Finally, using these parameters, the objective quantities can be computed. The detailed proofs involved to get observable representations of these parameters are in (Cohen et al., 2013). Intuitively the $\mathbf{Y}_*$ and $\mathbf{Z}$ are just projections of feature vectors of inside trees $\mathbf{T}_*$ and outside trees $\mathbf{O}$ to a lower dimensional space so that the inverse of entities in the observable representations exists.
Also, the minimum singular values of Ω^a are a measure of correlation between feature functions(ψ anf ϕ) and the hidden variables. This can be intuitively realized as $\mathbb{E}(\mathbf{X}, \mathbf{Y})$ is intimately related to the cross-covariance matrix.

3.3 Weighted Finite Automata

This section describes the spectral algorithm for Weighted Finite State Automata(WFA) described in (Balle et al., 2013). This algorithm is based on the relation between WFAs and Hankel matrices. Basically, the original parameters of WFA, which are not derivable from observations only, are replaced by surrogate parameters dependent upon low rank factorizations of Hankel matrices which depend only on observale strings. These surrogate parameters obtained by the spectral algorithm are used for estimating marginal probabilities of observables.

Hankel Matrix

Hankel matrices, $\mathbf{H}_f \in \mathbb{R}^{\Sigma^* \times \Sigma^*}$ are matrices encoding the function $f : \Sigma^* \to \mathbb{R}$. The rows and columns of a Hankel Matrix are prefixes and suffixes of strings. An immediate observation is that a Hankel matrix is infinite and represents the function of a string redundantly because a string can be formed by many combinations of prefixes and suffixes. The pragmatic setting employs sub-blocks of a Hankel matrix with a basis $B = (P, S)$ where P,S $\in \Sigma^*$. For spectral algorithms, closed basis are preferred,$B' = (P', S)$, where P'=P($\Sigma \cup \epsilon$). This results in $|\Sigma| + 1$ blocks of same size denoted by $H_\sigma \in \mathbb{R}^{P\sigma \times S} \, \forall \sigma \in (\Sigma \cup \epsilon)$, such that $H_\sigma(u, v) = H_f(u\sigma, v)$. Also, the algorithm works with only complete bases B such that $rank(H_B) = rank(H_f)$.
This rank of Hankel matrix is intimately related to the number of states(generally hidden) in a WFA.

WFA:Definition

A WFA over Σ with m states is represented as a triple $A = (\alpha_1, \alpha_\infty, \mathbf{A}_\sigma)$, where $\alpha_1, \alpha_\infty \in \mathbb{R}^m$ are initial and final weight vectors. $\mathbf{A}_\sigma \in \mathbb{R}^{m \times m}$ is the transition matix $\forall \sigma \in \Sigma$. Hence, the function f: $\Sigma^* \to \mathbb{R}$ for this WFA is defined as:

$$f_A(x) = \alpha_1^T A_{x_1} \dots A_{x_t} \alpha_\infty \tag{2}$$

(Carlyle and Paz, 1971) state an important result which says that f can be defined by a WFA iff rank(H_f) is finite, and in that case rank(H_f) is the minimal number of states of any WFA A such that f=f_A.
Also, a WFA is is invariant under the change of basis. A stochastic WFA is one which encodes a probability distribution function over strings and the algorithm focusses on such WFA. Generally, there are three interesting functions and their automata are related:

- $f(x) = \mathbb{P}[x]$ whose automata is $(\alpha_1, \alpha_\infty, \mathbf{A}_\sigma)$
- $f_p(x) = \mathbb{P}[x\Sigma^*]$ whose automata is $(\alpha_1, (\mathbf{I} - \mathbf{A})^{-1}\alpha_\infty, \mathbf{A}_\sigma)$
- $f_s(x) = \mathbb{P}[\Sigma^* x \Sigma^*]$ whose automata is $(\alpha_1(\mathbf{I} - \mathbf{A})^{-1}, (\mathbf{I} - \mathbf{A})^{-1}\alpha_\infty, \mathbf{A}_\sigma)$

where $\mathbf{A} = \sum_{\sigma \in \Sigma} \mathbf{A}_\sigma$ Due to this interelatedness, f can be realized by f_p or f_s and vice-versa. The rank of these three functions is equal.

Observable Representation

To formally represent the factorization of a Hankel Matrix n terms of WFA defined in the earlier sections: Let P $\in \mathbb{R}^{\Sigma^* \times m}$ encoding $\alpha_1^T \mathbf{A}_u \forall u \in \Sigma^*$ and S $\in \mathbb{R}^{m \times \Sigma^*}$ encoding $\mathbf{A}_u^T \alpha_\infty \forall v \in \Sigma^*$. We can see that $\mathbf{H_f} = \mathbf{PS}$. This is an important relation which shows that Hankel Matrices can factorize into WFA dependent entries. This is true for a sub-blocks of Hankel Matrices $\mathbf{H_B}$ with complete basis B=(P,S) inducing factorization $\mathbf{H}_B = \mathbf{P}_B \mathbf{S}_B$ and $\mathbf{H}_\sigma = \mathbf{P}_B \mathbf{A}_\sigma \mathbf{S}_B$.

Define $\mathbf{h}_{P,\epsilon} \in \mathbb{R}^P$ and $\mathbf{h}_{\epsilon,S} \in \mathbb{R}^S$ such that $\mathbf{h}_{P,\epsilon}(u) = f(u)$ and $\mathbf{h}_{\epsilon,S}(v) = f(v)$. If $\mathbf{h}$ and $\mathbf{H}$ are estimated, then SVD of $\mathbf{H}_\epsilon = (\mathbf{UA})\mathbf{V}^T$, and since $\mathbf{V}$ is orthogonal, $\mathbf{H}_\epsilon = (\mathbf{H}_\epsilon \mathbf{V})\mathbf{V}^T$.
If A is minimal then $\mathbf{H}_f = \mathbf{PS}$ is a rank factorization. Hence, a minimal WFA for f can be computed from factorization of a complete sub-block of Hankel Matrix. This result is the driving agent behind the following observable representation:

- $\alpha_\mathbf{1}^\mathbf{T} = \mathbf{h}_{\epsilon,S}^T \mathbf{V}$
- $\alpha_\infty = (\mathbf{H}_\epsilon \mathbf{V})^+ \mathbf{h}_{P,\epsilon}$
- $\mathbf{A}_\sigma = (\mathbf{H}_\epsilon \mathbf{V})^+ \mathbf{H}_\sigma \mathbf{V}$

Implementation

(Balle et al., 2013) performed experiments on PNFA. Number of latent WFA states n was identified by cross-validation. They experimented with two bases in their spectral algorithms: Σ basis and Top k prefix/suffix basis. Hence, the Hankel matrix sub-blocks were estimated over short sequences instead of full sequences. Therefore, the authors estimated $A_s = (\alpha_1(\mathbf{I} - \mathbf{A})^{-1}, (\mathbf{I} - \mathbf{A})^{-1}\alpha_\infty, \mathbf{A}_\sigma)$ from the spectral algorithm.
The spectral algorithms with these bases were compared with a Unigram Model, Bigram Model and EM Model on the task of predicting the next tag given a prefix of POS tags. EM and Spectral algorithms performed comparably and were better than unigram and bigram models. The runtime of spectral algorithm was found to be significantly less than that of EM algorithms.

3.4 Latent Tree Graphical Models

This section describes the spectral algorithm for estimating marginal probabilities over observed data given tree topologies involving latent variables, developed by (Song et al., 2011). The algorithm is fairly general but the paper considers the scenario where leaf nodes are always observed variables and internal nodes are latent variables. Also, as pointed out earlier, the number of states,n, of an observable variable is $\geq$ number of latent states,m. The fundamental claim made in their work is that after picking up an arbitrary root and then sorting the nodes in a topological order, the Conditional Probability Tables(CPT) between nodes and their parents are sufficient to characterize the joint or marginal distribution. Their representation requires a parent node to have atleast three children. If this is not the case, then sentinel children are introduced. If a node has more than three children then, algorithm is dependent only on any 3 consecutive children of the node, when traversed in a topological order. This representation requires tensors only upto 3-mode tensors which lead to a simple and practical spectral algorithm.

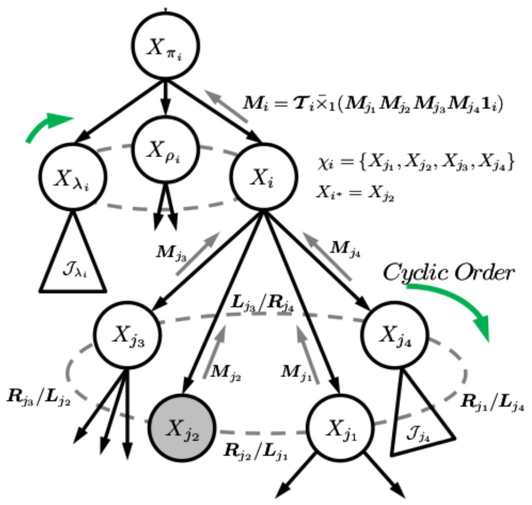


Figure 3: Part of a latent tree graphical model

In 3, the notations are: A node is denoted by X_i and its parent by X_{π_i}. The set of its children is dentoed by χ_i. The left and right siblings of X_i are denoted by X_{λ_i} and X_{ρ_i} respectively. Any observed variable in the subtree induced by X_i is denoted by X_{i^*}.
In a tree based algorithm if a root node is identified then the inference involves message passing. In the case of inference of observed variables which are all the leaf nodes of the tree, a specific tensor based message passing algorithm can be described.
The messages, denoted by $\mathbf{M} \in \mathbb{R}^{m \times m}$ are always diagonal. This uniformity allows easy message passing without taking care of dimensionality of messages during each propagation. Root of the tree is represented by $\mathbf{r} = \mathbb{P}(X_r) \in \mathbb{R}^m$. Each internal node is associated with a third order tensor $\mathcal{T}_i = \mathbb{P}(X_i|X_{\pi_i}) \in \mathbb{R}^{m \times m \times m}$, which is diagonal in its second and third mode. We get a diagonal message from this tensor by multiplying its first mode with $\mathbf{v} = \mathbb{P}(x_j|X_i) \in \mathbb{R}^m$ i.e. $\mathbf{M}_i = \mathcal{T} \times_1 \mathbf{v} = \mathbb{P}(x_j|X_{\pi_i})$. Each observed leaf node x_i is represented as a message $M_i = \mathbb{P}(x_i|X_{\pi_i})$ outgoing to its parent
With this representation, the outgoing message from an internal node X_i is computed as:
$\mathbf{M}_i = \mathcal{T}_i \times_1 (\mathbf{M}_{j_1} \dots \mathbf{M}_{j_J} \mathbf{1}_i)$, where each $\mathbf{M}_j$ is an incoming message from a child in χ_i. the above result is used to pass messages bottom up to the root in a recursive manner. The marginal probability is calculated at the root by following:

$$\mathcal{P}(x_1, \dots, x_O) = \mathbf{r}^T (\mathbf{M}_{j_1} \dots \mathbf{M}_{j_J} \mathbf{1}_r) \tag{3}$$

3.4.1 Spectral Algorithm

In the above settings, the CPT are generally not available and hence it is hard to estimate the original model parameters$\mathcal{T}$,$\mathbf{M}$ and $\mathbf{r} \in \theta$. Spectral algorithm focuses on recovering these parameters by some invertible transformations which can be computed from the observable data.
Each message $\mathbf{M}_j$ is transformed by two invertible matrices $\mathbf{L}_j$ and $\mathbf{R}_j$. The incoming message at X_i becomes:

$$\mathbf{M}_i = \mathcal{T}_i \times_1 (\mathbf{L_{j_1} L_{j_1}^{-1} M_{j_1} R_{j_1} L_{j_2}^{-1}} \dots \mathbf{L_{j_J}^{-1} M_{j_J} R_{j_J} R_{j_J}^{-1} 1_i}) \tag{4}$$

The outgoing message from X_i becomes:

$$\mathbf{M}_{\pi_i} = \mathcal{T}_{\pi_i} \times_1 (\dots \mathbf{L_i^{-1} M_i R_i} \dots \mathbf{1}_{\pi_i}) \tag{5}$$

Above two equations imply $\mathbf{R_{j_i} L_{j_{i+1}}}^{-1} = \mathbf{I}$. Hence $\mathbf{R}_j = \mathbf{L}_{\rho_j}$ and $\mathbf{L}_j = \mathbf{R}_{\lambda_j}$. Hence, the transformed parameters can be written as:

- $\mathcal{T}_i' = \mathcal{T}_i \times_1 \mathbf{L}_{j_1}^T \times_2 \mathbf{L}_i^{-1} \times_3 \mathbf{R}_i^T$
- $\mathbf{M}_j' = \mathbf{L}_j^{-1} \mathbf{M}_j \mathbf{R}_j$
- $\mathbf{1}_i' = \mathbf{R}_{j_J}^{-1} \mathbf{1}_i$
- $\mathbf{r}'^T = \mathbf{r}^T \mathbf{L}_{j_1}$

3.4.2 Observable Representation

Let $\mathbf{O}_{ij} = \mathbb{P}(X_i|X_j)$ be a conditional probability matrix, $\mathbf{U}_{j^*}$ be the matrix formed from m right singular vectors obtained from SVD of $\mathbb{P}(X_{\lambda_{j^*}}, X_{j^*})$. If we choose $\mathbf{L}_{j_1} = \mathbf{O}_{j_1^* i}^T \mathbf{U}_{j_1^*}$,$\mathbf{L}_i = \mathbf{O}_{i^* \pi_i}^T \mathbf{U}_{i^*}$ and $\mathbf{R}_i = \mathbf{O}_{\rho_i^* \pi_i}^T \mathbf{U}_{\rho_i^*}$, then the transformed parameters have an observable representation:

- $\mathcal{T}_i' = \hat{\mathbb{P}}(X_{j_1^*}, X_{\lambda_i^*}, X_{\rho_i^*}) \times_1 \hat{\mathbf{U}}_{j_1^*}^T \times_2 (\hat{\mathbb{P}}(X_{\lambda_i^*}, X_{i^*}) \hat{\mathbf{U}}_{i^*})^+ \times_3 \hat{\mathbf{U}}_{\rho_i^*}$
- $\mathbf{M}_i' = (\hat{\mathbb{P}}(X_{\lambda_i^*}, X_i) \hat{\mathbf{U}}_i)^+ \hat{\mathbb{P}}(X_{\lambda_i^*}, x_i, X_{\rho_i^*}) \hat{\mathbf{U}}_{\rho_i^*}$
- $\mathbf{1}_i' = (\hat{\mathbb{P}}(X_{j_J^*}, X_{\rho_{j_J}^*}) \hat{\mathbf{U}}_{\rho_{j_J}^*})^+ \hat{\mathbb{P}}(X_{j_J^*})$
- $\mathbf{r}'^T = \hat{\mathbb{P}}(X_{j_1^*})^T \hat{U}_{j_1^*}$

These transformed parameters are then used to compute marginal probability over the observed variables according to the message passing algorithm.

Implementation

This technique was implemented over 4 different tree topologies. The comparisons were made with the EM algorithm and Chow-Liu Tree algorithms. The performance of the spectral methods was comparable to the the EM algorithms. It varied with the size of the training samples. However Spectral algorithms were reported to be much faster than EM algorithms. Also, their performance was much better than Chow-liu's algorithm but this comparison favoured spectral algorithms as they work with prespecified topologies.

4 Spectral Algorithms for Parameter Estimation

Much work on Spectral Learning algorithms has ignored the estimation of original model parameters and has just focussed on distributution of observables.(Anandkumar et al., 2012) focus on formulating higher order tensor decomposition to facilitate the estimation of parameters. A significant difference in their approach and all the other models discussed in this paper up till now is that they employ decomposition of tensors having an order higher than 2 to estimate the parameters.

4.1 Intuition behind Tensor decomposition

The tensor decomposition algorithm are inspired from the matrix factorization algortihms and work with 'generalized' rayleigh quotient for tensor decomposition instead of the rayleigh quotient used to factorize matrices.
For a matrix M, the desired factorization is of the form $\mathbf{M} = \mathbf{V \Lambda V}^T$ where $\mathbf{V} = [v_1|..|v_k]$ is a matrix made

out of orthonormal vectors and $\Lambda = diag(\lambda_1, ...\lambda_k)$ is a diagonal matrix of corresponding eigenvalues. These forms when represented as a tensor product look like: $\sum_{i=1}^{k} \lambda_i v_i^{\otimes 2}$. The intuition of generalization to higher order tensor decomposition is that the form of the desired decomposition should be an extension of its 2nd-order(matrix) counterpart. Hence, for 3-order tensor, the desired decomposition is of the form: $\sum_{i=1}^{k} \lambda_i v_i^{\otimes 3}$. Here $\otimes$ symbol denotes the outer product. Hence an outer product of $v_i \in \mathbb{R}^n$ and $v_j \in \mathbb{R}^n$ is a $n \times n$ matrix(2nd order tensor). So, $v^{\otimes p} = v \otimes v.. \otimes v$(p times) results in a p-order tensor.

4.2 Method of Moments for LVMs

Various latent variable models with varying levels of complexity are discussed. In their algorithms, the common underlying approach employs observable moments of various orders to yield a representation which results in tensors formed by outer products of parameters(latent variables). These observable tensors are then decomposed to estimate the parameters. This survey discusses the simplest model called 'exachangable single topic model', formulated by them, but it must be noted that similar approaches can be used to estimate parameters of more complex models like HMMs, LDA etc.
An 'exachangable single topic model' is a simple bag of words model where the permutations of the words in the document don't affect the distribution of words in the document. There is just one latent variable(topic) h, for the whole document which can take k values. The observable words x_is are conditionally independent of each other given h. The size of the vocabular is d and hence each word is represented as a d-dimesnsional vector $\in \mathbb{R}^d$. Let there be l words in a document. Given the topic h, the words are generated according to a discrete distribution specified by the probability vector μ_h. In the notation $e_1..e_d$ is the basis for $\mathbb{R}^d$ i.e. they are one hot vectors representing the word in the vocabulary. Another parameter is the topic probability: $Pr[h = j] = w_j \forall j \in [1 \ldots k]$. With this parametrization,

$$\mathbb{E}[x_t | h = j] = \sum_{i=1}^{d} [\mu_j]_i e_i = \mu_j$$

and

$$\mathbb{E}[x_1 \otimes x_2 | h = j] = \mathbb{E}[x_1 | h = j] \otimes \mathbb{E}[x_2 | h = j] = \mu_j \otimes \mu_j \forall j \in [1 \ldots k]$$

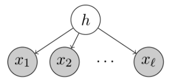


Figure 4: Exchangable single topic model

In this simple case, it is easy to use 2nd and 3rd order moments to get a tensorial representation of model parameters.

$$M_2 = x_1 \otimes x_2 = \sum_{i=1}^{k} \mu_i \otimes \mu_i$$

$$M_3 = x_1 \otimes x_2 \otimes x_3 = \sum_{i=1}^{k} \mu_i \otimes \mu_i \otimes \mu_i$$

We can see from the above equation observable moments computed using x values can be represented as tensors involving models parameters w and μ. Another fact to be noted is that due to exchangibility in this model, we need not restrict ourselves to the first few bigrams,trigrams as was the case in the HMM formulation by (Hsu et al., 2012). Incorporating all the bigrams and trigrams intuitively should yield more

reliable parameter estimates.
For more complicated models, the higher order moments need to be manipulated to get the parametric tensorial representations.

4.3 Parameter Estimation

Here, we focus on the parameter estimation of the simple model described above using orthogonal tensor decomposition. The moments M_2 and M_3 are used for estimating μs and ws. Again, the natural seperability condition requiring μs to be linearly independent should be satisfied for any decomposiitons performed on the tensors. Another condition that must be satisfied is that ws are strictly positive. These two conditions boil down to requiring M_2 to be positive definite and have rank $\geq$ k.
In the estimation procedure, first a matrix W is determined such that $M_2(W, W) = W^T MW = I$. Here W can be $UD^{-1/2}$, where U is the matrix for orthonormal vectors of M_2 and D is the diagonal matrix of M_2's eigenvalues. Now if we let:

$$\hat{\mu} = \sqrt{w_i} W^T \mu_i$$

then we can see from the definition of W that $\sum_{i=1}^{k} \hat{\mu}_i \hat{\mu}_i^T = \mathbf{I}$ This shows that this definition of $\hat{\mu}$ makes μ_is orthonormal vectors. Using W and $\hat{\mu}$, $M_3(W, W, W) = \sum_{i=1}^{k} w_i (W^T \mu_i)^{\otimes 3} = \sum_{i=1}^{k} \frac{1}{\sqrt{w_i}} \hat{\mu}_i^{\otimes 3}$.

Using $M_3(W, W, W)$ computed from the observables and equating it to the form above we can conclude that eigenvectors of $M_3(W, W, W)$ are $\hat{\mu_i}$s and the eigenvalues are $\frac{1}{w_i}$s.
The above example recovers all the parameter using second and third order moments of the observables. For more complex models, there will be changes in the abovementioned procedure according to the manipulations done on the higher order moments for the parametric tensorial forms.

5 Sample Complexities

In the spectral HMM models proposed by (Hsu et al., 2012), for there to be a difference of ϵ between the true joint distribution and the estimated joint distribution with probability $1 - \eta$,

$$N \geq C(fract\epsilon)^2 (\frac{k}{\sigma_k(O)^2 \sigma_k(P_{2,1})^4} + \frac{kn_0(\epsilon_0)}{\sigma_k(O)^2 \sigma_k(P_{2,1})^2}) log \frac{1}{\eta}$$

where N is the number of training examples,σ_k is the kth largest eigenvalue of a matrix. In this case, k is the number of hidden states possible. and $\epsilon_0 = \frac{\sigma_k(O)\sigma_k(P_{2,1})}{4t\sqrt{k}}$

In latent variable PCFGs, for the bound

$$1 - \epsilon \leq fracp(r_1 \hat{\ldots} r_C) p(r_1 \ldots r_C) \leq 1 + \epsilon$$

the sample complexity is

$$N \geq \frac{128k^2}{(\sqrt[2C+1]{1+\epsilon} - 1)^2 \Lambda^2 \sigma^4} log(\frac{2kR}{\eta})$$

where R is the total number of rules and Λ, σ are the coefficients for $\hat{U}$ and $\hat{V}$ of the PCFG.

In the latent tree graphical models, for there to be a difference of ϵ between the true joint distribution and the estimated joint distribution with probability $1 - \eta$,

$$N \geq O(\frac{(d_{max}k)^{2l+1}}{min_{i \neq j}(\sigma_k(P[X_i, X_j])^4) min_i(\sigma_k(O_{i,\pi_i})^2)\epsilon^2}) log \frac{\mathbb{O}}{eta}$$

where l is the length of the chain of hidden variables.

In all the three cases we can see that estimation gets harder as k(number of latent states) increases.

6 Summary

Various models and their spectral algorithms have been discussed in this paper. All the spectral approaches have, at the core, a unifying principle which involves use of multiple low rank factorizations of matrices/tensors which is helpful for estimation based upon latent cariable models.
The HMM and WFSA models discusssed are very similar in terms of their aims but the inputs to the algorithms differ slightly and the framework too is quite different. In the HMM model, only intial unigrams, bigrams and trigrams are considered to build the model and estimate marginal probabilities or predict the next entity in the sequence. WFSA model considers observable distribution of prefixes and suffixes as input and also works on partial sequences.
Latent Variable PCFG models are more complicated than the HMM models and also account for feature vectors in the inside outside algorithm.
Latent tree graphical models are fairly general models and a tensor based spectral method was developed for the cases when the tree topology is already known.
Finally, a general method involving high order moments was also discussed which offers to estimate parameters of LVMs too instead of just the posterior probabilities.

7 Conclusion

This survey described the spectral approaches to model Latent Variable Models, which are very prevalant in Structured Prediciton problems. It discusses about the models which make use of the low-rank factorization of probability matrices instead of trying to optimize over a distribution. Various models for which spectral algorithms have been developed, were discussed. In general, not many approaches using spectral methods have focussed on parameter estimation but this survey also discusses about some work on this problem.
These approaches provide a local-minima free, consistent and fast estimation algorithms. But, they have some drawbacks too. They do not aim to optimize the likelihood or some other such statistical measure. Also, the factorizations needed for observable representations are not intuitive at all, which makes the development of algorithms for general graphical models difficult.

References

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. (2012). Tensor decompositions for learning latent variable models. *arXiv preprint arXiv:1210.7559*.

Balle, B., Carreras, X., Luque, F. M., and Quattoni, A. (2013). Spectral learning of weighted automata.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Carlyle, J. W. and Paz, A. (1971). Realizations by stochastic finite automata. *Journal of Computer and System Sciences*, 5(1):26–40.

Cohen, S. B., Stratos, K., Collins, M., Foster, D. P., and Ungar, L. (2013). Experiments with spectral learning of latent-variable pcfgs. In *Proceedings of NAACL-HLT*, pages 148–157.

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

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098.

Hsu, D., Kakade, S. M., and Zhang, T. (2012). A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480.

Matsuzaki, T., Miyao, Y., and Tsujii, J. (2005). Probabilistic cfg with latent annotations. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 75–82. Association for Computational Linguistics.

Song, L., Xing, E. P., and Parikh, A. P. (2011). A spectral algorithm for latent tree graphical models. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1065–1072.