Spectral Learning Methods for Latent Variables

Kartik Goyal
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
kartikgo@cs.cmu.edu

1 Introduction

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, 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.

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. This survey focusses on identification of common theoretical considerations and analysis of such spectral algorithms that have been developed for various LVM models like Hidden Markov Models, Latent PCFG trees, LDA etc. 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.

1.1 Intuition behind Spectral Algorithms

Let us 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.

1.2 General Algorithm

In general, 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.
Much of the work has focussed on estimating the marginal or conditional probabilities of observables and don't concern themselves with the estimation of latent parameters. In this setting, a typical spectral algorithm for 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 quantity in which we are interested(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.

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.
There have also been algorithms which aim to estimate the latent parameters as well. In these 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.

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. 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

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,

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

3 Discussion and Results

Several algorithms that focus on estimating the marginal or conditional probabilities of observed data, have been proposed to model various LVMs like Hidden Markov Models (Hsu et al., 2012), Latent variable tree graphical models(Song et al., 2011), latent variable PCFGs Cohen et al. (2013), and weighted Finite state automata (Balle et al., 2013). All these algorithms make different assumptions and their algorithms are slightly different from each other but they all follow the same underlying procedure summarized above. This section discusses the parametrization of some of these models, and also delves into the sample complexity analyses for these models, which are motivated by the

same underlying principle and proof structure.
Also, work by (Anandkumar et al., 2012) that focuses on estimating the latent parameters, is also discussed. This is implemented fairly differently from the algorithms mentioned above and it uses second and third order moments to estimate latent parameters.

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 $m \leq n$.

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:

$$Pr[x_1, \ldots, x_t] = \tilde{\mathbf{1}}_{\mathbf{m}}^{\mathbf{T}} \mathbf{A}_{\mathbf{x_t}} \ldots \mathbf{A}_{\mathbf{x_1}} \tilde{\pi}$$

Observables

In their theoretical model, they analyse an algorithm that use initial few observations of the sequence and ignore the rest. They are marginal probabilities of observation singletons($\mathbf{P}_1 \in \mathbb{R}^n$),pairs($\mathbf{P}_{2,1} \in \mathbb{R}^{n \times n}$) and triples($P_{3,x,1} \in \mathbb{R}^{n \times n}$). 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. Based upon these quantities surrogate parameters are defined as follows:

- $\tilde{\mathbf{b}}_{\mathbf{1}} = \mathbf{U}^{\mathbf{T}} \mathbf{P}_{\mathbf{1}} = (\mathbf{U}^{\mathbf{T}} \mathbf{O} \vec{\pi})$
- $\tilde{\mathbf{b}}_{\infty} = (\mathbf{P}_{\mathbf{2,1}}^{\mathbf{T}} \mathbf{U})^{+} \mathbf{P}_{\mathbf{1}} = \tilde{\mathbf{1}}_{\mathbf{m}} (\mathbf{U}^{\mathbf{T}}))^{-1}$
- $\mathbf{B}_{\mathbf{x}} = (\mathbf{U}^{\mathbf{T}} \mathbf{P}_{\mathbf{3,x,1}})(\mathbf{U}^{\mathbf{T}} \mathbf{P}_{\mathbf{2,1}})^{+} \; \forall x \in [n]$
- $\tilde{\mathbf{b}}_{\mathbf{t+1}} = \frac{\mathbf{B}_{\mathbf{x_t}} \tilde{\mathbf{b}_t}}{\tilde{\mathbf{b}_{\infty}}^{\mathbf{T}} \mathbf{B}_{\mathbf{x_t}} \tilde{\mathbf{b}_t}}$

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:

- $\mathbf{Pr}[x_1, \ldots, x_t] = \mathbf{b}_{\infty} \mathbf{B}_{\mathbf{x_t}} \ldots \mathbf{B}_{\mathbf{x_1}} \mathbf{b_1}$
- $\mathbf{Pr}[x_t | x_{1:t-1}] = \frac{\mathbf{b}_{\infty}^{\mathbf{T}} \mathbf{B}_{\mathbf{x_t}} \tilde{\mathbf{b}_t}}{\sum_x \tilde{\mathbf{b}_{\infty}}^{\mathbf{T}} \mathbf{B}_{\mathbf{x_t}} \tilde{\mathbf{b}_t}}$

3.2 Latent Tree Graphical Models

This more general setting, developed by (Song et al., 2011), borrows a lot of techniques from the HMM model described above. 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. A node is denoted by X_i and its parent by X_{π_i}. The set of its children is denoted 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:

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

3.2.1 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:

$$\mathbf{M}_i = \mathcal{T}_i \times_1 (\mathbf{L_{j_1}}\mathbf{L_{j_1}^{-1}}\mathbf{M_{j_1}}\mathbf{R_{j_1}}\mathbf{L_{j_2}^{-1}} \dots \mathbf{L_{j_J}^{-1}}\mathbf{M_{j_J}}\mathbf{R_{j_J}}\mathbf{R_{j_J}^{-1}}\mathbf{1_i}) \tag{2}$$

The outgoing message from X_i becomes:

$$\mathbf{M}_{\pi_i} = \mathcal{T}_{\pi_i} \times_1 (\dots \mathbf{L_i^{-1}}\mathbf{M_i}\mathbf{R_i} \dots \mathbf{1}_{\pi_i}) \tag{3}$$

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

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

3.2.2 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 an observable representation of the transformed parameters can be found out, which can be used to compute marginal probability over the observed variables according to the message passing algorithm.

3.3 Sample Complexities

The sample complexity analyses of models like HMM and latent tree structures have a similar proof structure, which depends upon the number of latent states, the length of the sequence(chain of variables)and the eigenvalues of relevant matrices corresponding to the number of latent states in the model. In both the models, the number of observations required for restricting the error in calculating the true joint distribution to ϵ with probability $1-\eta$, are estimated.
In the spectral HMM model,

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

where N is the number of training examples, l is the length of the chain of hidden variables, $n_0(\epsilon)$ is the minimum number of observations that account for about $1-\epsilon$ of the total probability mass, 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 the latent tree graphical models,

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

where d_{max} is the maximum degree of a node(In HMM it is 2).
In both the cases, we can see that estimation gets harder as k(number of latent states) increases, as not only is the complexity polynomial in terms of k but σ_k also gets smaller with k which increases the complexity of the models. Although, as we discus in the next section, the proof structure of both the models complexities is very similar, we observe that the complexity of general latent tree structures is exponential in l compared to the polynomial dependence on l in case of HMMs, which hints towards a possibility of further tightening the bounds for the latent tree models. Both the models are freed from the dependence on number of observable states,n by assuming that frequency of observation symbols follow a certain distribution.

3.4 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.

3.4.1 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 $\mathbf{M} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ 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.

3.4.2 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: $Pr[h = j] = w_j \forall j \in [1 \dots k]$. With this parametrization,

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

and

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

The tensor representation of model parameters is:

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

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

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.

3.4.3 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 $M_2(W, W) = W^T M W = I$. 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 above-mentioned procedure according to the manipulations done on the higher order moments for the parametric tensor forms.

4 Proof outlines for the Complexity Analyses

The proof structures of the models which estimate joint probabilities of variables, follow a general approach described in this section. The examples are used from the HMM model but they can be extended to other models like latent trees easily. The steps in obtaining the complexity bounds are:

- Estimate the concentration bounds of the observed quantities. These are the sampling errors from the true distribution of observables like $\mathbf{P_{1,2}}$ in HMM. The Frobenius norm of matrix errors is bounded. This bound is obtained by applying McDiarmid's inequality and realizing that expected difference is approximately $\sqrt{N}$.
- Next, all the bounds for eigenvalues of the relevant matrices are found.
- This step is a major step which uses the above two results of concentration bounds and eigenvalue bounds, results from Matrix perturbation theory and triangle inequality to estimate the bounds over transformed quantities like B_x, b_∞, b_1 in HMM.
- Finally, the propagation error in the model is bounded by usng Holder's inequality which results in the fact that the errors due to multiplying matrices in a sequence roughly accumulate additively and not exponentially.
- Finally, using the results from the above four steps, the bound over joint probability is found.

5 Conclusion

This survey described the spectral approaches to model Latent Variable Models which have at the core, a unifying principle which involves use of multiple low rank factorizations of matrices/tensors which is very different from the idea of optimizing over a distribution or likelihood. These approaches provide a local-minima free, consistent and fast estimation algorithms. This paper discusses a approaches that only focus on estimating joint probabilities of variables as well as the approaches that attempt to estimate the parameters related to latent variables too.
These approaches have some drawbacks too. They do not aim to optimize the likelihood or some other such explicit statistical measure and the factorizations needed for observable representations are not intuitive, which makes the development of algorithms for general graphical models difficult. Moreover, the sample complexity often depends on the eigenvalues of the model quantities which can only be computed after the implementation of the algorithm.

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.