# Hierarchical Tensor Decomposition of Latent Tree Graphical Models

**Le Song, Haesun Park** {LSONG,HPARK}@CC.GATECH.EDU
College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA
**Mariya Ishteva** MARIYA.ISHTEVA@VUB.AC.BE
ELEC, Vrije Universiteit Brussel, 1050 Brussels, Belgium
**Ankur Parikh, Eric Xing** {APPARIKH,EPXING}@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

## Abstract

We approach the problem of estimating the parameters of a latent tree graphical model from a hierarchical tensor decomposition point of view. In this new view, the marginal probability table of the observed variables is treated as a tensor, and we show that: (*i*) the latent variables induce low rank structures in various matricizations of the tensor; (*ii*) this collection of low rank matricizations induces a hierarchical low rank decomposition of the tensor. We further derive an optimization problem for estimating (alternative) parameters of a latent tree graphical model, allowing us to represent the marginal probability table of the observed variables in a compact and robust way. The optimization problem aims to find the best hierarchical low rank approximation of a tensor in Frobenius norm.

For correctly specified latent tree graphical models, we show that a global optimum of the optimization problem can be obtained via a recursive decomposition algorithm. This algorithm recovers previous spectral algorithms for hidden Markov models (Hsu et al., 2009; Foster et al., 2012) and latent tree graphical models (Parikh et al., 2011; Song et al., 2011) as special cases, elucidating the global objective these algorithms are optimizing. For misspecified latent tree graphical models, we derive a novel decomposition based on our framework, and provide approximation guarantee and computational complexity analysis. In both synthetic and real world data, this new estimator significantly improves over the state-of-the-art.

*Proceedings of the 30$^{th}$ International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. 

## 1. Introduction

Latent tree graphical models capture rich probabilistic dependencies among random variables and find applications in various domains, such as modeling dynamics, clustering, and topic modeling (Rabiner & Juang, 1986; Clark, 1990; Hoff et al., 2002; Blei et al., 2003). Recently, there is an increasing interest in designing spectral algorithms for estimating the parameters of latent variable models (Hsu et al., 2009; Parikh et al., 2011; Song et al., 2011; Foster et al., 2012). Compared to Expectation-Maximization (EM) algorithms (Dempster et al., 1977) traditionally used for the same task, the advantages of spectral algorithms are their computational efficiency and good theoretical guarantees. Unlike EM, these spectral algorithms recover a set of alternative parameters (rather than the original parameters) which consistently estimate only the marginal distributions of observed variables.

Previous spectral algorithms are carefully constructed based on the linear algebraic properties of latent tree graphical models. It is however unclear what objective function these algorithms are optimizing and whether they are robust when latent tree graphical models are misspecified. Recently, Balle et al. (2012) provided some partial answers to these questions by formulating the problem in terms of a regularized *local* loss minimization and presenting their results in the setting of weighted automata. However, it is still unclear whether spectral algorithms can be interpreted from a *global* loss minimization point of view as typically used in iterative algorithms such as EM, and how to further understand these algorithms from basic linear algebraic point of view, using concepts such as low rank approximations. Therefore, the goal of this paper is to provide new insight to these questions.

Our first contribution is deriving a global objective function and the optimization space for spectral algorithms. More specifically, we approach the problem of estimating the parameters of latent tree graphical

models from a hierarchical tensor decomposition point of view. In this new view, the marginal probability tables of the observed variables in latent tree graphical models are treated as tensors, and we show that the space of tensors associated with latent tree graphical models has the following two properties: (*i*) various matricizations of the tensor according to the edges of the tree are low rank matrices; (*ii*) this collection of low rank matricizations induces a hierarchical low rank decomposition for the tensors. Overall, the optimization problem aims to minimize the Frobenius norm of the difference between the original tensor and a new tensor from the space of hierarchical low rank tensors.

Our second contribution is showing that previous spectral algorithms for hidden Markov models (Hsu et al., 2009; Foster et al., 2012) and latent tree graphical models (Parikh et al., 2011; Song et al., 2011) are special cases of the proposed framework, which elucidates the global objective these algorithms are optimizing for. Essentially, these algorithms recursively apply a low rank matrix decomposition result to solve the optimization problem. When the latent tree models are correctly specified, these algorithms find a global optimum of the optimization problem.

When the latent tree models are misspecified, previous spectral algorithms are no longer optimal. Our third contribution is deriving a better decomposition algorithm for these cases, based on our hierarchical low rank tensor decomposition framework, and providing some theoretical analysis. In both synthetic and real world data, the new algorithm significantly improves over previous state-of-the-art spectral algorithms.

## 2. Background

**Latent tree graphical models (LTGM).** We focus on discrete latent variable models whose conditional independence structures are *undirected* trees. We use uppercase letters to denote random variables ($X_i$) and lowercase letters for their instantiations ($x_i$). A latent tree graphical model defines a joint probability distribution over a set of $O$ observed variables $\{X_1, \ldots, X_O\}$ and a set of $H$ hidden variables $\{X_{O+1}, \ldots, X_{O+H}\}$. For simplicity, we assume that (**I**) all observed variables are leaves having $n$ states, $\{1, \ldots, n\}$, and all hidden variables have $k$ states, $\{1, \ldots, k\}$, with $k \leq n$.

The joint distribution of all variables in a latent tree graphical model is fully characterized by a set of conditional probability tables (CPTs). More specifically, we can arbitrary select a node in the tree as root, and sort the nodes in the tree in topological order. Then the set of CPTs between nodes and their parents $P(X_i|X_{\pi_i})$ (the root node $X_r$ has no parent, so $P(X_r|X_{\pi_r}) = P(X_r)$) are sufficient to characterize the joint distribution, $P(x_1, \ldots, x_{O+H}) = \prod_{i=1}^{O+H} P(x_i|x_{\pi_i})$. The marginal distribution of the observed variables can be obtained by summing out the latent ones,

$$P(x_1, \ldots, x_O) = \sum_{x_{O+1}} \cdots \sum_{x_{O+H}} \prod_{i=1}^{O+H} P(x_i|x_{\pi_i}).$$

Latent tree graphical models allow complex distributions over observed variables (*e.g.*, clique models) to be expressed in terms of more tractable joint models over the augmented variable space. This is a significant saving in model parametrization.

**Latent tree graphical models as tensors.** We view the marginal distribution $P(X_1, \ldots, X_O)$ of a latent tree model as a tensor $\mathcal{P}$, each variable corresponding to one mode of the tensor. The ordering of the modes is not essential so we simply label them using the corresponding random variables.

We can reshape (unfold) a tensor into a matrix by grouping some of its modes into rows and the remaining ones into columns. The resulting matrix has exactly the same entries as the original tensor but they are reordered. Let $\mathscr{O} = \{X_1, \ldots, X_O\}$ be the set of modes and $\mathscr{I}_1$ and $\mathscr{I}_2$ be two disjoint subsets with $\mathscr{O} = \mathscr{I}_1 \cup \mathscr{I}_2$. Similarly to the Matlab function,

$$P_{\mathscr{I}_1;\mathscr{I}_2} = reshape\left(P(X_1, \ldots, X_O), \mathscr{I}_1\right)$$

denotes a matricization of $P(X_1, \ldots, X_O)$ for which variables corresponding to $\mathscr{I}_1$ are mapped to rows and those corresponding to $\mathscr{I}_2$ are mapped to columns. Each row of the resulting matrix corresponds to an assignment of the variables in $\mathscr{I}_1$. For instance, $P_{\{X_2\};\{X_1,X_3\}} = reshape(P(X_1, X_2, X_3), \{X_2\})$, and for simplicity we also use $P_{\{2\};\{1,3\}}$ to denote $P_{\{X_2\};\{X_1,X_3\}}$. We arrange the row indexes such that the values of variables with lower index change faster than those with higher index. We similarly arrange the column indexes. We will overload the reshape operation to deal with matrices. For instance, we may reshape $P_{\{1,2,3\};\{4\}}$ into $P_{\{1,2\};\{3,4\}}$ by shifting variable $X_3$ from rows to columns, *i.e.*, $P_{\{1,2\};\{3,4\}} = reshape(P_{\{1,2,3\};\{4\}}, \{X_1, X_2\})$.

## 3. Hierarchical Low Rank Structure

We will show that the latent tree structure $\mathcal{T}$ induces a hierarchical low rank structure in $P(X_1, \ldots, X_O)$. We will reshape this tensor into a collection of matrices, each of which corresponding to an edge in the latent tree. We will show that (*i*) although the sizes of these matrices can be exponential in the number of variables, the ranks of these matrices cannot exceed the number of states $k$ of the hidden variables; (*ii*) the low rank structures of this collection of matricizations further induce a hierarchical decomposition of the tensor.

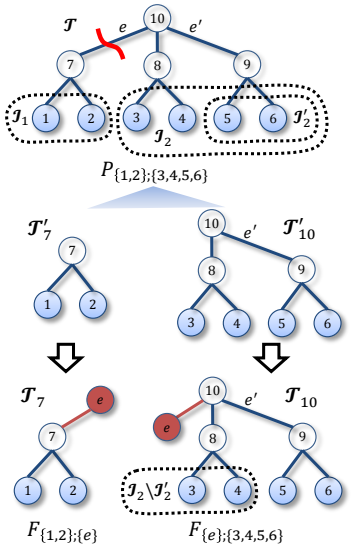



*Figure 1.* Latent tree graphical model with 6 observed variables $\{X_1, \ldots, X_6\}$ (shaded) and 4 hidden variables $\{X_7, \ldots, X_{10}\}$ (transparent). We divide the tree $\mathcal{T}$ into subtrees $\mathcal{T}'_7$ and $\mathcal{T}'_{10}$ by cutting primary edge $e = (X_7, X_{10})$. By adding dummy variables $X_e$ to each subtree, we obtain $\mathcal{T}_7$ and $\mathcal{T}_{10}$, and the corresponding factors $F_{\{1,2\};\{e\}}$ and $F_{\{e\};\{3,4,5,6\}}$. The *primary* edges are the ones present in $\mathcal{T}$.

### 3.1. Low Rank Matricizations of Tensors

Each edge in the latent tree corresponds to a pair of variables $(X_s, X_t)$ which induces a partition of the observed variables into two groups, $\mathscr{I}_1$ and $\mathscr{I}_2$ (such that $\mathscr{O} = \mathscr{I}_1 \cup \mathscr{I}_2$ and $\emptyset = \mathscr{I}_1 \cap \mathscr{I}_2$). One can imagine splitting the latent tree into two subtrees by cutting the edge. One group of variables reside in the first subtree, and the other group in the second subtree. If we unfold the tensor according to this partitioning, then

**Theorem 1** *Under condition* **I**, $\text{rank}(P_{\mathscr{I}_1;\mathscr{I}_2}) \leq k$.

**Proof** Due to the conditional independence structure induced by the latent tree, $P(x_1, \ldots, x_O) = \sum_{x_s} \sum_{x_t} P(\mathscr{I}_1|x_s) P(x_s, x_t) P(\mathscr{I}_2|x_t)$, which can be written in a matrix form as

$$P_{\mathscr{I}_1;\mathscr{I}_2} = P_{\mathscr{I}_1|\{s\}} P_{\{s\};\{t\}} P^{\top}_{\mathscr{I}_2|\{t\}}, \tag{1}$$

where $P_{\mathscr{I}_1|\{s\}} = \mathit{reshape}(P(\mathscr{I}_1|X_s), \mathscr{I}_1)$, $P_{\mathscr{I}_2|\{t\}} = \mathit{reshape}(P(\mathscr{I}_2|X_t), \mathscr{I}_2)$ and $P_{\{s\};\{t\}} = P(X_s, X_t)$. $P_{\{s\};\{t\}}$ is a $k \times k$ matrix, so its rank is at most $k$. Since the rank of a product of matrices cannot exceed the rank of any of its factors, $\text{rank}(P_{\mathscr{I}_1;\mathscr{I}_2}) \leq k$. ■

Theorem 1 implies that although the dimensions of the matricizations are large, their rank is bounded by $k$. For instance, $P_{\{\mathscr{I}_1\};\{\mathscr{I}_2\}}$ has size $n^{|\mathscr{I}_1|} \times n^{|\mathscr{I}_2|}$, exponential in the number of observed variables, but its rank is at most $k$. Essentially, given a latent tree graphical model, we obtain a collection of low rank matricizations $\{P_{\mathscr{I}_1;\mathscr{I}_2}\}$ of the tensor $P(X_1, \ldots, X_O)$, each corresponding to an edge $(X_s, X_t)$ of the tree. More interestingly, the low rank structures of the matricizations also imply that the tensor can be decomposed hierarchically as we see later.

**Algorithm 1** *decompose*$(\mathcal{P}, \mathcal{T}, \mathcal{E}, k)$

**Input:** tensor $\mathcal{P}$, tree $\mathcal{T}$, set of primary edges $\mathcal{E}$, rank $k$.
**Output:** factors of a hierarchical rank $k$ decomposition of $\mathcal{P}$, according to tree $\mathcal{T}$ and primary edges $\mathcal{E}$.

1: Pick a *primary* edge $e = (X_s, X_t) \subset \mathcal{E}$ and:
 – partition the tensor modes into $\{\mathscr{I}_1, \mathscr{I}_2\}$,
 – matricize the tensor $\mathcal{P}$ to $P_{\mathscr{I}_1;\mathscr{I}_2}$,
 – split the tree $\mathcal{T}$ into two subtrees $\mathcal{T}'_s$ and $\mathcal{T}'_t$,
 – split $\mathcal{E} \backslash e$ into $\mathcal{E}_s$ and $\mathcal{E}_t$, w.r.t. the subtrees.
2: Decompose $P_{\mathscr{I}_1;\mathscr{I}_2}$ as

$$P_{\mathscr{I}_1;\mathscr{I}_2} = F_{\mathscr{I}_1;\{e\}} \; M \; F^{\top}_{\mathscr{I}_2;\{e\}}, \tag{2}$$

where $\text{rank}(M) = k$.
3: According to (2):
 – associate the columns of the factors $F_{\mathscr{I}_1;\{e\}}$ and $F_{\mathscr{I}_2;\{e\}}$ with a new label $X_e$,
 – introduce a dummy "observed" leaf $X_e$ to each subtree ($\mathcal{T}'_s$ and $\mathcal{T}'_t$),
 – join $X_e$ with $X_s$ in $\mathcal{T}'_s$ to form $\mathcal{T}_s$; join $X_e$ with $X_t$ in $\mathcal{T}'_t$ to form $\mathcal{T}_t$,
 – reshape $F_{\mathscr{I}_1;\{e\}}$ and $F_{\mathscr{I}_2;\{e\}}$ back to tensors $\mathcal{F}_s$ and $\mathcal{F}_t$, respectively, each mode corresponding to either an observed or a dummy variable.
4: Call *decompose*$(\mathcal{F}_s, \mathcal{T}_s, \mathcal{E}_s, k)$ if $\mathcal{E}_s \neq \emptyset$; call *decompose*$(\mathcal{F}_t, \mathcal{T}_t, \mathcal{E}_t, k)$ if $\mathcal{E}_t \neq \emptyset$.

### 3.2. Hierarchical Low Rank Decomposition

We say that a tensor $\mathcal{P}$ has hierarchical rank $k$ according to a tree $\mathcal{T}$ with leaves corresponding to the modes of $\mathcal{P}$, if it can be *exactly* decomposed according to Algorithm 1. The decomposition is carried out recursively according to $\mathcal{T}$ and its primary edges $\mathcal{E}$ (see Fig. 1 for notation). The end result of the hierarchical (recursive) decomposition is a collection of matrices and 3rd order tensors (or factors with 3 indexes). By reshaping and combining these factors in reverse order of the recursion, we can obtain the original tensor $\mathcal{P}$. Alternatively, one can think that each entry in $\mathcal{P}$ is obtained by a sequence of sum and product of factors. That is, $P(x_1, \ldots, x_O) = \sum_{x_e} \prod_{i=1}^{O} F(x_i, \cdot) \prod \mathcal{F}(\cdot, \cdot, \cdot)$, where the unspecified indexes in $F$ and $\mathcal{F}$ correspond to "dummy" variables, and the summation ranges over all "dummy" variables $X_e$. We denote $\mathcal{H}(\mathcal{T}, k)$ the class of tensors $\mathcal{P}$ admitting hierarchical rank $k$ decomposition according to tree $\mathcal{T}$.[1] Similar decompositions have also been proposed in tensor community, but not for latent variable models (Grasedyck, 2010; Oseledets, 2011).

[1] One can readily generalize this notation to decompositions where different factors can have different ranks.

### 3.3. Low Rank Matricizations Induce Hierarchical Low Rank Decomposition

Next, we show that if all matricizations of a tensor $\mathcal{P}$ according to a tree $\mathcal{T}$ have rank at most $k$, then $\mathcal{P}$ admits a hierarchical rank $k$ decomposition, *i.e.*, $\mathcal{P} \in \mathcal{H}(\mathcal{T}, k)$. We note that this property applies to a general tensor $\mathcal{P}$ where the tensor modes $\{X_1, \ldots, X_O\}$ are organized hierarchically into a tree structure $\mathcal{T}$. A latent tree graphical model is a special case. In general, the low rank factors $F$ and $\mathcal{F}$ in the hierarchical decomposition do not have an interpretation as CPTs. More specifically, we have the following theorem

**Theorem 2** *Let $\mathcal{P}$ be a tensor and $\mathcal{T}$ be a tree. If $\text{rank}(P_{\mathscr{I}_1;\mathscr{I}_2}) \leq k$ for every matricization $P_{\mathscr{I}_1;\mathscr{I}_2}$ of $\mathcal{P}$ according to an edge of $\mathcal{T}$, then $\mathcal{P}$ admits a hierarchical rank $k$ decomposition, i.e., $\mathcal{P} \in \mathcal{H}(\mathcal{T}, k)$.*

**Proof** Let the modes of the tensor be labeled as $X_1, \ldots, X_O$ and $\{\mathscr{I}_1, \mathscr{I}_2\}$ be a partition of the modes according to an edge $e$ of $\mathcal{T}$. Since $\text{rank}(P_{\mathscr{I}_1;\mathscr{I}_2}) \leq k$, it admits a rank $k$ decomposition $P_{\mathscr{I}_1;\mathscr{I}_2} = UV^\top$ (with $U$ and $V$ having $k$ columns), or in index form

$$P(x_1, \ldots, x_O) = \sum\nolimits_{x_e} U(x_{\mathscr{I}_1}, x_e) V(x_{\mathscr{I}_2}, x_e).$$

The matrix $V$ can be expressed as $V = P_{\mathscr{I}_1;\mathscr{I}_2}^\top (U^\dagger)^\top = P_{\mathscr{I}_1;\mathscr{I}_2}^\top W$ or in index form

$$V(x_{\mathscr{I}_2}, x_e) = \sum\nolimits_{x_{\mathscr{I}_1}} P(x_{\mathscr{I}_1}, x_{\mathscr{I}_2}) W(x_{\mathscr{I}_1}, x_e).$$

Now the matrix $V$ can be reshaped into a $(|\mathscr{I}_2|+1)$-th order tensor $\mathcal{V}$ with a new tree $\mathcal{T}_t$ and a dummy variable $X_e$. We will consider its unfolding according to an edge $e' = (X_{s'}, X_{t'})$ in this new tree (and the associated partition of tensor modes $\{\mathscr{I}_1', \mathscr{I}_2'\}$)

$$V_{\mathscr{I}_1', \mathscr{I}_2'} = reshape(V, \mathscr{I}_1')$$

and show that $\text{rank}(V_{\mathscr{I}_1', \mathscr{I}_2'}) \leq k$ holds. Suppose $\mathscr{I}_2' \subset \mathscr{I}_2$ and its complement $\bar{\mathscr{I}}_2' = \{1, \ldots, O\} \setminus \mathscr{I}_2'$. Then matricizing the original tensor $\mathcal{P}$ according to edge $e'$, we have $P_{\bar{\mathscr{I}}_2';\mathscr{I}_2'} = reshape(\mathcal{P}, \bar{\mathscr{I}}_2')$ or in index form (see Fig. 1 for notation)

$$P(x_1, \ldots, x_O) = \sum\nolimits_{x_{e'}} F(x_{\bar{\mathscr{I}}_2'}, x_{e'}) G(x_{e'}, x_{\mathscr{I}_2'}),$$

which also has rank $k$. Using this, we obtain

$$\begin{aligned} V(x_{\mathscr{I}_2}, x_e) &= \sum\nolimits_{x_{\mathscr{I}_1}} W(x_{\mathscr{I}_1}, x_e) P(x_{\mathscr{I}_1}, x_{\mathscr{I}_2}) \\ &= \sum_{x_{\mathscr{I}_1}} \sum_{x_{e'}} W(x_{\mathscr{I}_1}, x_e) F(x_{\bar{\mathscr{I}}_2'}, x_{e'}) G(x_{e'}, x_{\mathscr{I}_2'}). \\ &= \sum\nolimits_{x_{e'}} R(\{x_e, x_{\mathscr{I}_2 \setminus \mathscr{I}_2'}\}, x_{e'}) G(x_{e'}, x_{\mathscr{I}_2'}) \end{aligned}$$

and $R(x_e, x_{\mathscr{I}_2 \setminus \mathscr{I}_2'}, x_{e'}) = \sum_{\mathscr{I}_1} W(x_{\mathscr{I}_1}, x_e) F(x_{\bar{\mathscr{I}}_2'}, x_{e'})$. Now row and column indexes of $V_{\mathscr{I}_1';\mathscr{I}_2'}$ are separated and $\text{rank}(V_{\mathscr{I}_1', \mathscr{I}_2'}) \leq k$. The process can be carried out recursively until obtaining a hierarchical rank $k$ decomposition. ■

Theorem 2 suggests that the low rank constraints on the matricizations of the original tensor $\mathcal{P}$ induce a hierarchical low rank decomposition of $\mathcal{P}$. This result allows us to define the space of tensors $\mathcal{H}(\mathcal{T}, k)$ using the matricizations rather than the recursive decomposition in Algorithm 1 or the factorization form.

## 4. Optimization Problem

Now we can write out an optimization problem which recovers previous spectral algorithms as special cases and suggests new algorithms. Given a tensor $\mathcal{P}$ which might not admit a hierarchical low rank decomposition as $\mathcal{H}(\mathcal{T}, k)$, we seek the closest (in Frobenius norm) hierarchical low rank tensor in $\mathcal{H}(\mathcal{T}, k)$. That is,

$$\min_{\mathcal{Q}} \quad \|\mathcal{P} - \mathcal{Q}\|_F^2, \qquad s.t. \;\; \mathcal{Q} \in \mathcal{H}(\mathcal{T}, k), \tag{3}$$

where the constraint is equivalent to $\text{rank}(Q_{\mathscr{I}_1;\mathscr{I}_2}) \leq k$ for all matricizations of $\mathcal{Q}$ according to edges of tree $\mathcal{T}$. In general, the optimization problem is not convex.

Learning the parameters of a latent tree graphical model is a special case of optimization problem (3). In this case, $\mathcal{P}$ is the joint probability tensor of the observed variables, and we seek a hierarchical rank $k$ decomposition $\mathcal{Q} \in \mathcal{H}(\mathcal{T}, k)$ with tree structure $\mathcal{T}$. The question is whether we can design an efficient algorithm for carrying out this hierarchical decomposition. Naively applying existing low rank decomposition techniques to the unfoldings of the tensor will result in algorithms with exponential computational cost ($O(n^O)$ or even higher) since such algorithms typically operate on all entries of the input matrix. Therefore, the goal is to develop efficient low rank decomposition algorithm that exploits the structure of the problem.

When $\mathcal{P}$ itself admits a hierarchical low rank decomposition, $\mathcal{P} \in \mathcal{H}(\mathcal{T}, k)$, or $\mathcal{P}$ is indeed generated from a (correctly specified) latent tree graphical model, we can achieve $\|\mathcal{P} - \mathcal{Q}\|_F^2 = 0$. The solution is however usually not unique in terms of the factors in $\sum_{x_e} \prod_{l=1}^O F(x_l, \cdot) \prod \mathcal{F}(\cdot, \cdot, \cdot)$, since we can easily generate different factors by applying invertible transformations. Many previous spectral algorithms are special cases of this framework (§4.1).

An interesting question arises when $\mathcal{P}$ itself does not admit a hierarchical low rank $k'$ decomposition $\mathcal{H}(\mathcal{T}, k')$, but we want to obtain a $\mathcal{Q} \in \mathcal{H}(\mathcal{T}, k')$ which best approximates it (misspecified model with $k'$ smaller than the true rank). In this case, the optimization is more difficult. Previous spectral algorithms in general cannot achieve the best objective and lack approximation guarantees. In §4.2 we propose a new algorithm to cope with this situation which has approximation guarantees and low computational complexity.

### 4.1. Correctly Specified Models

Having the correct $k$, we can derive hierarchical low rank decompositions with zero approximation error. We first prove a matrix equality key to the decomposition. It will be applied recursively according to the latent tree structure, each time reducing the size of the latent tree (and the joint probability tables). This recursive algorithm also provides a new view of previous spectral algorithms for latent tree graphical models.

**Theorem 3** *Let matrix $P \in \mathbb{R}^{l\times m}$ have rank $k$. Let $A \in \mathbb{R}^{l\times k}$ and $B \in \mathbb{R}^{m\times k}$ be such that $rank(A^\top P) = k$ and $rank(PB) = k$. Then $P = PB\ (A^\top PB)^{-1}\ A^\top P$.*

**Proof** Let the singular value decomposition of $P$ be $P = \begin{pmatrix} U & U_\perp \end{pmatrix} \begin{pmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} V & V_\perp \end{pmatrix}^\top = U\Sigma V^\top$, where $U \in \mathbb{R}^{l\times k}$ and $V \in \mathbb{R}^{m\times k}$ have orthonormal columns, $U_\perp$ and $V_\perp$ are their orthogonal complements and $\Sigma \in \mathbb{R}^{k\times k}$ is a diagonal nonsingular matrix. Then $A$ and $B$ can be represented $A = UC + U_\perp D$ and $B = VE + V_\perp F$ respectively, where $C, E \in \mathbb{R}^{k\times k}$. Then we obtain

$$A^\top P = C^\top \Sigma V^\top, \quad PB = U\Sigma E, \quad A^\top PB = C^\top \Sigma E.$$

Note that since $rank(A^\top P) = rank(V^\top B) = k$, then $C$ and $E$ are nonsingular. Finally, we prove the claim

$$\begin{aligned} PB\ (A^\top PB)^{-1}\ A^\top P &= U\Sigma E\ (C^\top \Sigma E)^{-1}\ C^\top \Sigma V^\top \\ &= U\Sigma E E^{-1}\Sigma^{-1} C^{-\top} C^\top \Sigma V^\top \end{aligned}$$

noting that the r.h.s equals $U\Sigma V^\top = P$. ■

We can recursively apply Theorem 3 according to Algorithm 1. We only need to let $P = P_{\{\mathscr{I}_1\};\{\mathscr{I}_2\}}$, and replace the r.h.s. of equation (2) by $PB\ (A^\top PB)^{-1}\ A^\top P$, with

$$F_{\mathscr{I}_1;\{e\}} = PB, \quad F_{\mathscr{I}_2;\{e\}} = A^\top P, \quad M = (A^\top PB)^{-1},$$

Furthermore, we will choose $A$ and $B$ such that $PB$ and $A^\top P$ are easy to compute. In particular, when $P$ is the reshaped probability matrix, $A$ and $B$ are chosen to marginalize (or sum) out certain variables.

**The latent tree model in Fig. 1.** First, let us examine the case $n = k$, and all pairwise marginal distributions are invertible. Then we can reshape the joint probability table $P(X_1, \ldots, X_6)$ according to edge $e = (X_7, X_{10})$ and decompose the unfolding to

$$\underbrace{P_{\{1,2\};\{3,4,5,6\}}}_{P} = \underbrace{P_{\{1,2\};\{3\}}}_{PB}\ \underbrace{P^{-1}_{\{2\};\{3\}}}_{(A^\top PB)^{-1}}\ \underbrace{P_{\{2\};\{3,4,5,6\}}}_{A^\top P},$$

where $A = I_n \otimes \mathbf{1}_n$ sums out variable $X_1$, and $B = \mathbf{1}_n \otimes \mathbf{1}_n \otimes \mathbf{1}_n \otimes I_n$ sums out variables $X_4$, $X_5$ and $X_6$ ($I_n$ is the $n\times n$ identity matrix, $\mathbf{1}_n$ is a vector of all ones of length $n$). We call the variables appearing in the the middle matrix ($X_2$ and $X_3$) the linker variables. The choice of linker variables is arbitrary as long as they reside in different sides of edge $e$. Carrying out such decomposition recursively,

$$P_{\{2,3,4\};\{5,6\}} = P_{\{2,3,4\};\{5\}}\ P^{-1}_{\{4\};\{5\}}\ P_{\{4\};\{5,6\}}$$

where $A = I_n \otimes \mathbf{1}_n \otimes \mathbf{1}_n$, and $B = \mathbf{1}_n \otimes I_n$. And then

$$P_{\{3,4\};\{2,5\}} = P_{\{3,4\};\{2\}}\ P^{-1}_{\{4\};\{2\}}\ P_{\{4\};\{2,5\}}$$

where $A = I_n \otimes \mathbf{1}_n$ and $B = \mathbf{1}_n \otimes I_n$. In the end only second and third order factors are left, and no further reduction of the order of the tensor can be made.

When $k < n$, the middle matrices for the linker variables are no longer invertible. For instance, $P_{\{2\};\{3\}}$ has size $n \times n$, but $P(x_2, x_3) = \sum_{x_{10}} P(x_2|x_{10})P(x_{10})P(x_3|x_{10})$ which means $rank(P_{\{2\};\{3\}}) \le k < n$. In this case, we can use a slightly modified $A$ and $B$ matrices and Theorem 3 still applies. More specifically, we will introduce matrices $U_2$ and $V_3$ with orthonormal columns such that $U_2^\top P_{\{2\};\{3\}} V_3$ is invertible, and this leads to the following decomposition of $P_{\{1,2\};\{3,4,5,6\}}$

$$\underbrace{P_{\{1,2\};\{3\}}V_3}_{PB}\ \underbrace{(U_2^\top P_{\{2\};\{3\}}V_3)^{-1}}_{(A^\top PB)^{-1}}\ \underbrace{U_2^\top P_{\{2\};\{3,4,5,6\}}}_{A^\top P},$$

where $A = (I_n \otimes \mathbf{1}_n)U_2$ and $B = (\mathbf{1}_n \otimes \mathbf{1}_n \otimes \mathbf{1}_n \otimes I_n)V_3$ perform marginalization and projection simultaneously. A natural choice for $U_2$ and $V_3$ is provided by the singular value decomposition of $P_{\{2\};\{3\}}$, *i.e.*, $P_{\{2\};\{3\}} = U_2\Sigma V_3^\top$. Likewise, we can carry out such decomposition recursively with $U$ and $V$ matrices introduced in each decomposition.

A nice feature of the above recursive decomposition algorithm is that it never needs to access all entries in the tensor and it only works on small linker matrices (*e.g.*, inverting $P_{\{2\};\{3\}}$). This hierarchical decomposition also provides latent tree graphical models with a representation using only marginal distributions of triplets of observed variables. Furthermore, many previous spectral algorithms become special cases.

**Parikh et al. (2011); Song et al. (2011)** proposed spectral algorithms for general latent tree graphical models. The main difference between their approach and our framework is in the linker matrix. For instance, $P_{\{1,2\};\{3,4,5,6\}}$ is decomposed as $P_{\{1,2\};\{3\}}V_3(P_{\{2\};\{3\}}V_3)^\dagger P_{\{2\};\{3,4,5,6\}}$ in their method where $^\dagger$ is the pseudo inverse. We note that

$$(P_{\{2\};\{3\}}V_3)^\dagger = (U_2^\top P_{\{2\};\{3\}}V_3)^{-1}U_2 \qquad (4)$$

which is the $k \le n$ case under our framework.

**Hsu et al. (2009); Foster et al. (2012)** derived spectral algorithms for hidden Markov models which are special latent tree graphical models. The building block for the reduced dimension model of Foster et al. (2012) coincides with the decomposition from

Theorem 3 although they derived their model from a very different perspective. Furthermore, they show that their model is equivalent to that of Hsu et al. (2009) by making use of the relation in (4).

### 4.2. Misspecified Models

The algorithms we discussed in §4.1 assume that $\mathcal{P}$ itself admits a hierarchical rank $k$ decomposition, *i.e.*, $\mathcal{P} \in \mathcal{H}(\mathcal{T}, k)$. In practice, we do not know the exact hierarchical rank $k$ for $\mathcal{P}$, and we want to obtain an approximate latent tree graphical model by using a $k'$ different from the true $k$ (usually, $k' < k$). In this case, it becomes difficult to obtain a global optimum of the optimization problem in (3). And these spectral algorithms no longer have performance guarantees. In this section, we will consider a particular type of misspecification in models: $\mathcal{P} \in \mathcal{H}(\mathcal{T}, k)$, but we supply the algorithms with a $k'$ such that $k' < k$, where we design a new algorithm with provable guarantees.

When $k' < k$, the matrix decomposition result used in previous spectral algorithms in general produces only an approximation, *i.e.*, $P \approx PB\ (A^\top PB)^{-1}\ A^\top P$. Furthermore, given $A$ and $B$ (or given the edge where we split the tree), the middle linker matrix $M = (A^\top PB)^{-1}$ is no longer the best choice. Instead, we will use a new linker matrix (Yu & Schuurmans, 2011)

$$M^* = \operatorname*{argmin}_{\operatorname{rank}(M)\le k'} \left\| P - PB\ M\ A^\top P \right\|_F^2, \qquad (5)$$

which has a closed form expression as

$$M^* = (PB)^\dagger \left( U_B U_B^\top\ P\ V_A V_A^\top \right)_{(k')} \left( A^\top P \right)^\dagger, \qquad (6)$$

where $A^\top P = U_A \Sigma_A V_A^\top$, $PB = U_B \Sigma_B V_B^\top$ and $(\cdot)_{(k')}$ denotes the truncation of its matrix argument at $k'$-th singular value. When $k' = k$, the new decomposition reduces to the decomposition in §4.1.

We can apply $P \approx (PB) M^* (A^\top P)$ recursively according to Algorithm 1. Again, we only need to let $P = P_{\{\mathscr{I}_1\};\{\mathscr{I}_2\}}$, and replace the right hand side of equation (2) by $(PB)\ M^*\ (A^\top P)$ with

$$F_{\mathscr{I}_1;\{e\}} = PB, \quad F_{\mathscr{I}_2;\{e\}} = A^\top P, \quad M = M^*.$$

Here we no longer have an exact decomposition in each step of the recursion, and the final hierarchical decomposition is an approximation to the original tensor.

**For the model in Fig. 1,** its joint probability tensor $P(X_1, \ldots, X_6)$ can be decomposed as

$$\underbrace{P_{\{1,2\};\{3,4,5,6\}}}_{P} \approx \underbrace{P_{\{1,2\};\{3\}}}_{PB}\ M^*_{\{2\};\{3\}}\ \underbrace{P_{\{2\};\{3,4,5,6\}}}_{A^\top P},$$

where $A$ and $B$ are set as before, and the matrix $M^*_{\{2\};\{3\}}$ has rank $k'$ and it minimizes $\|P_{\{1,2\};\{3,4,5,6\}} - P_{\{1,2\};\{3\}} M P_{\{2\};\{3,4,5,6\}}\|_F^2$. Such decomposition are then carried out recursively on $P_{\{2\};\{3,4,5,6\}}$ and so on until only second and third order tensors are left. In the end, we obtain a set of low order tensors, and by combining them backwards we obtain an approximation to the original tensor.

## 5. Analysis of the Decomposition for Misspecified Models

We provide further analysis of the properties of the hierarchical decomposition in Algorithm 1 for misspecified models, including approximation guarantees, a sketch analysis of the sample complexity and computational complexity.

### 5.1. Approximation Guarantee

Applying Algorithm 1 with equation (6) in the misspecified case does not provide a global optimal solution. It only constructs a particular $\mathcal{Q} \in \mathcal{H}(\mathcal{T}, k')$ based on $\mathcal{P}$. Nonetheless, we can provide an approximation guarantee which was considered in previous spectral algorithms.

**Theorem 4** *Let the matricizations of $\mathcal{P} \in \mathcal{H}(\mathcal{T}, k)$ according to edge $e_i$ of the latent tree $\mathcal{T}$ be $P_i$, and its best rank $k'$ approximation error be $\epsilon_i = \min_{\operatorname{rank}(R)\le k'} \|P_i - R\|_F^2$. Let $\mathcal{Q} \in \mathcal{H}(\mathcal{T}, k')$ be a hierarchical rank $k'$ decomposition using Algorithm 1 and equation (6). Assume* $\operatorname{rank}(PB) = \operatorname{rank}(A^\top P) = \operatorname{rank}(P)$ *in all recursive applications of (6), then*

$$\|\mathcal{P} - \mathcal{Q}\|_F^2 \le d\epsilon, \qquad (7)$$

*where $\epsilon = \max\{\epsilon_i, \forall e_i\}$ and $d$ is the number of edges.*

**Proof** Suppose in each iteration we choose the edge to split in such a way that only one subtree needs to be further decomposed. We denote $P_L = PB = F_{\mathscr{I}_1;\{e\}}$ and $P_R = A^\top P = F_{\mathscr{I}_2;\{e\}}$ for quantities in equation (2). This means that we further decompose $P_R$ as $\widetilde{P}_R$, but not $P_L$. Then the approximation error $\|\mathcal{P} - \mathcal{Q}\|_F^2$ can be bounded as

$$\|P - P_L\ M^*\ \widetilde{P}_R\|_F^2 = \|P - P_L M^* (\widetilde{P}_R - P_R + P_R)\|_F^2$$

$$= \|P - P_L M^* P_R\|_F^2 + \|P_L M^* (\widetilde{P}_R - P_R)\|_F^2 \qquad (8)$$

$$\le \epsilon + \|P_L M^* (\widetilde{P}_R - P_R)\|_F^2 \ \le\ \epsilon + \ldots = d\epsilon. \qquad (9)$$

where in (8), we used $(P - P_L M^* P_R)^\top P_L M^* = \mathbf{0}$; in (9), we used $\|P_L M^* (\widetilde{P}_R - P_R)\|_F^2 \le \epsilon$ and applied induction on the edges in the latent tree. ∎

Furthermore, under similar conditions, the new hierarchical decomposition is close to optimal.

**Theorem 5** *Let $\mathcal{Q}^* = \operatorname{argmin}_{\mathcal{Q}\in\mathcal{H}(\mathcal{T},k')} \|\mathcal{P} - \mathcal{Q}\|_F$ and $\mathcal{Q} \in \mathcal{H}(\mathcal{T}, k')$ be obtained from Algorithm 1 based on equation (6). Then $\|\mathcal{Q} - \mathcal{P}\|_F^2 \le d\,\|\mathcal{Q}^* - \mathcal{P}\|_F^2$ where $d$ is the number of edges in the latent tree $\mathcal{T}$.*

**Proof** In Theorem 4, $\epsilon_i := \min_{\text{rank}(R)=k'} \|P_i - R\|_F^2$ is the error for the best rank $k'$ approximation to unfolding $P_i$. However, $\mathcal{Q}^*$ minimizes the same objective but with more constraints. Hence $\epsilon_i \leq \epsilon^* = \|\mathcal{Q}^* - \mathcal{P}\|_F^2$ for all $e_i$. Using Theorem 4, this proves the claim. ■

### 5.2. Computational Complexity

When we are provided finite samples only, let $\widehat{P}$ be the finite sample estimate of $P$ needed in the decomposition in Algorithm 1. The major computations of the algorithm are repeatedly computing $M^*$ in equation (5). First, we observe

$$\begin{aligned}&\text{argmin}_{rank(M)\leq k'} \|\widehat{P} - (\widehat{P}B)M(A^\top \widehat{P})\|_F \\ =\ &\text{argmin}_{rank(M)\leq k'} \|Q_B^\top \widehat{P} Q_A - R_B M R_A^\top\|_F\end{aligned}$$

where $\widehat{P}B = Q_B R_B$ and $A^\top \widehat{P} = R_A^\top Q_A^\top$ are QR-factorizations, and $R_B, R_A \in \mathbb{R}^{n\times n}$ are small square matrices. We note that QR-factorization is generally faster than SVD, and after QR-factorization, we can then compute SVD for much smaller matrices in equation (6). Second, matrix $B\widehat{P}$ (similarly $A^\top \widehat{P}$ and $\widehat{P}$) is the unfolding of certain joint probability table, and is extremely sparse: the number of nonzero entries is not larger than the number of data points and much smaller than the size of the matrix $\widehat{P}$ (which can be $O(n^O)$). We can exploit this fact and use sparse matrix operations. For instance, the QR-factorization needs to work only on the nonzero entries. This gives us a $Q_B$ which has just a small number of nonzero rows (no larger than the number of data points). Although this new algorithm is more expensive than the previous spectral algorithms, it is still much faster than the EM algorithm as we observed in the experiments.

### 5.3. Sample Complexity

Given $m$ samples, we only have a finite sample estimate $\widehat{\mathcal{P}}$ of the the joint probability tensor $\mathcal{P}$. Then we hierarchically decompose $\widehat{\mathcal{P}}$ into $\widehat{\mathcal{Q}}$ using Algorithm 1. The Frobenius norm difference between $\widehat{\mathcal{Q}}$ and the true $\mathcal{P}$ can be decomposed into two terms

$$\|\widehat{\mathcal{Q}} - \mathcal{P}\|_F \leq \underbrace{\|\widehat{\mathcal{Q}} - \mathcal{Q}\|_F}_{\text{estimation error}} + \underbrace{\|\mathcal{Q} - \mathcal{P}\|_F}_{\text{approximation error}}, \quad (10)$$

where $\mathcal{Q}$ is the hierarchical decomposition of $\mathcal{P}$ using Algorithm 1. The result in Theorem 4 shows that the approximation error is bounded by $\sqrt{d\epsilon}$. Furthermore $\epsilon$ is determined by the tail of the singular values of the matricizations $P_i$ of $\mathcal{P}$. That is $\epsilon = \max\{\sum_{j>k'} \sigma_j(P_i), \forall e_i\}$ where $\sigma_j(\cdot)$ returns the $j$-th singular value of its argument. We can see that when $k' = k$, $\epsilon = 0$ and the approximation error is 0. In general, when $k' < k$, the number of nonzero singular value of $P_i$ can be exponential in its size, which is about $O(n^O)$. If the singular values decays very fast, then the approximation error can still be small.

Estimation error can be bounded in two main steps: first, the empirical estimators for all joint probability matrices needed in the hierarchical decomposition have nice finite sample guarantees, typically of order $O(m^{-1/2})$; second, the estimation errors can accumulate as we combine lower order tensor components to form the final joint probability tensor. For the moment, we do not yet have a complete sample complexity analysis, which deserves a full treatment in a separate paper. In sketch, we expand the error for the first iteration of the recursive decomposition ($P_L$, $P_R$ and $\widetilde{P}_R$ are defined similarly as in Theorem 4, and the versions with hat are finite sample counterparts):

$$\begin{aligned}&\|\mathcal{Q} - \widehat{\mathcal{Q}}\|_F \leq \|\widehat{P}_L\ \widehat{M}^*\ \widehat{\widetilde{P}}_R - P_L\ M^*\ \widetilde{P}_R\|_F \\ \leq\ &\|P_L\ M^*\ (\widehat{\widetilde{P}}_R - \widetilde{P}_R)\|_F + \|(\widehat{P}_L\ \widehat{M}^* - P_L\ M^*)\ \widetilde{P}_R\|_F \\ &+ \|(\widehat{P}_L\ \widehat{M}^* - P_L\ M^*)\ (\widehat{\widetilde{P}}_R - \widetilde{P}_R)\|_F \\ \precsim\ &\frac{\sigma_1(P)}{\sigma_k(P_R)}\|\widehat{\widetilde{P}}_R - \widetilde{P}_R\|_F + \frac{\sigma_1(P)}{\sigma_k(P_R)}\|\widehat{P} - P\|_F\end{aligned}$$

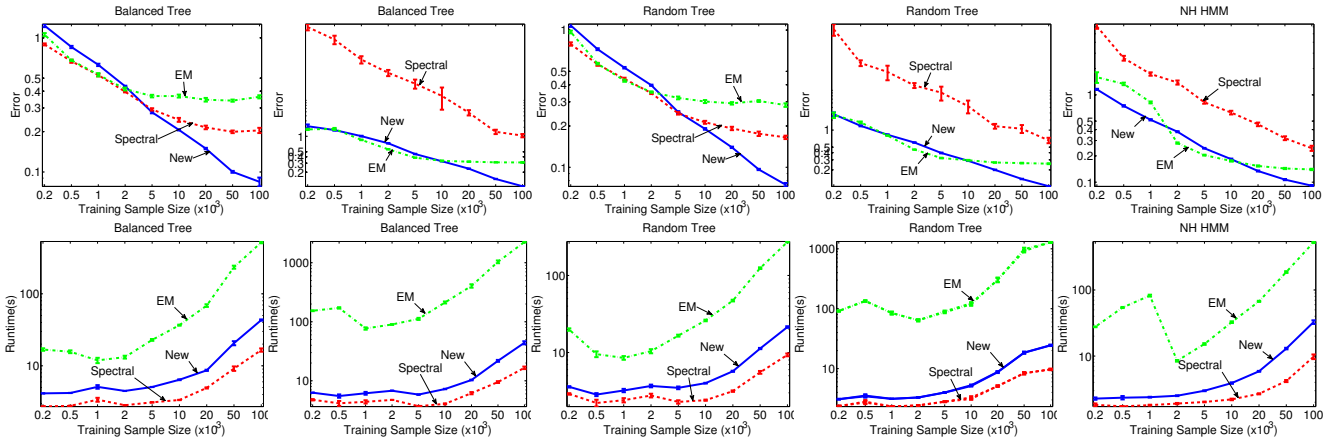
where in the last inequality, we have ignored higher order error terms and use $\precsim$ to denote "approximately larger". The occurrence of the $k$-th singular value $P_R$ in the denominator is due to the pseudo-inverse term in $M^*$ (see equation (6)). Then the error analysis for the subsequent steps of recursion can be carried out on $\|\widehat{\widetilde{P}}_R - \widetilde{P}_R\|_F$. Based on this sketch, the error will accumulate exponentially with respect to the number of recursion, resulting in a sample complexity of $O(c^O m^{-1/2})$ where $c$ is some constant dependent on the singular values. However, in our experiments, we did not observe such exponential degradation.

## 6. Experiments

We compare the new algorithm with EM (Dempster et al., 1977) and spectral algorithms of Hsu et al. (2009); Parikh et al. (2011).

**Synthetic data.** We generate synthetic data from latent tree models with different topologies: non-homogeneous hidden Markov models (NH HMM), random and balanced binary latent tree graphical models. We experimented with a variety of observed states $n$, hidden states $k$ and approximation state $k'$.

For an experiment on a given tree type with $N$ training points, we randomly generate 10 sets of model parameters and sample $N$ training points and 1000 test points for each parameter set. For EM, we learn the CPTs (with 5 restarts) based on the training points and the true latent tree topology, and then perform inference on test points using message passing. Conver-



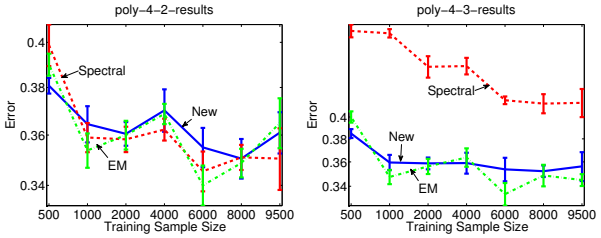

(a) $\{32, 6, 4, 2\}$ (b) $\{32, 10, 7, 5\}$ (c) $\{25, 6, 4, 2\}$ (d) $\{25, 10, 7, 5\}$ (e) $\{20, 10, 7, 5\}$

*Figure 2.* Error (top plots) and training time (bottom plots) in estimating the marginal probability tables of observed variables in latent tree graphical models when the models are misspecified. The setting of each experiment is denoted using a quadruple $\{O, n, k, k'\}$.

gence for EM is determined by measuring the change in the log likelihood at iteration $t$ (denoted by $f(t)$) over the average: $\frac{|f(t)-f(t-1)|}{\text{avg}(f(t),f(t-1))} \leq 10^{-4}$. We measure the performance of estimating the joint probability of observed variables using $\epsilon = \frac{|\widehat{P}(x_1,\ldots,x_O)-P(x_1,\ldots,x_O)|}{P(x_1,\ldots,x_O)}$, and we vary the training sample size $N$ from 200 to 100,000, and plot the test error for inference in Fig. 2.

Fig. 2 shows that our new algorithm performs the best when the sample sizes grows. Although EM performs the best for small training sizes, its performance levels off when the sample sizes go beyond 5,000 and our new algorithm overtakes EM. Furthermore, in our experiments when both $n$ and $k$ are reasonably large, previous spectral algorithms become less stable. In terms of training time, the new algorithm is more expensive than previous spectral algorithms, but still much faster than the EM algorithm.

**Genome sequence data.** We next consider the task of predicting *poly*$(A)$ motifs in DNA sequences (Kalkatawi et al., 2012) and consider the $AATAAA$ variant of the motif. This is a binary sequence classification problem (either the sequence has the motif or it doesn't). For each sequence, we take a contiguous subsequence of 100 nucleotides, which we model as a non-homogeneous hidden Markov model with $n = 4$ observed states($A$, $T$, $C$, $G$). We then vary the training set size from 500 to 9500, while the test set size is fixed to 800. We compare our approach with EM and spectral for both $k' = 2$ and $k' = 3$. The classification results are shown in Fig. 3. For $k' = 2$, all algorithms are relatively comparable. For $k' = 3$, our new approach and EM also perform comparably. However, spectral algorithm performs considerably worse as the number of hidden states increases.




(a) $\{O, n, k'\} = \{100, 4, 2\}$ (b) $\{O, n, k'\} = \{100, 4, 3\}$

*Figure 3.* Results on poly(A) dataset. $O$: the number of observed variables, $n$: the number of states for observed variables, $k'$: number of hidden states supplied to the algorithms. *New*: refers to our new hierarchical decomposition algorithm using (6).

## 7. Conclusions

We approach the problem of estimating the parameters of a latent tree graphical model from a hierarchical low rank tensor decomposition point of view. Based on this new view, we derive the global optimization problem underlying many existing spectral algorithms for latent tree graphical models. We show that these existing algorithms obtain a global optimum when the models are correctly specified. However, when the models are misspecified, these spectral algorithms are no longer optimal. Based on our framework, we derived a new decomposition algorithm with provable approximation guarantee and show the empirical advantages of our approach. In future, we will perform statistical analysis of this new algorithm and investigate potentially better algorithm.

**Acknowledgments:** Research supported by Georgia Tech Startup Fund; NSF IIS1218749; US government; ERC Grant 258581; Belgian Network DYSCO - IAP VII; NSF Graduate Fellowship 0946825; NIH 1R01GM093156.

## References


Balle, Borja, Quattoni, Ariadna, and Carreras, Xavier. Local loss optimization in operator models: A new insight into spectral learning. In *Proceedings of the International Conference on Machine Learning*, 2012.

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

Clark, A. Inference of haplotypes from PCR-amplified samples of diploid populations. *Molecular Biology and Evolution*, 7(2):111–122, 1990.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–22, 1977.

Foster, D.P., Rodu, J., and Ungar, L.H. Spectral dimensionality reduction for hmms. *Arxiv preprint arXiv:1203.6130*, 2012.

Grasedyck, Lars. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

Hoff, Peter D., Raftery, Adrian E., and Handcock, Mark S. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

Hsu, D., Kakade, S., and Zhang, T. A spectral algorithm for learning hidden markov models. In *Proc. Annual Conf. Computational Learning Theory*, 2009.

Kalkatawi, M., Rangkuti, F., Schramm, M., Jankovic, B.R., Kamau, A., Chowdhary, R., Archer, J.A.C., and Bajic, V.B. Dragon polya spotter: predictor of poly (a) motifs within human genomic dna sequences. *Bioinformatics*, 28(1):127–129, 2012.

Oseledets, IV. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

Parikh, A., Song, L., and Xing, E. P. A spectral algorithm for latent tree graphical models. In *Proceedings of the International Conference on Machine Learning*, 2011.

Rabiner, L. R. and Juang, B. H. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3 (1):4–16, January 1986.

Song, L., Parikh, A., and Xing, E.P. Kernel embeddings of latent tree graphical models. In *Advances in Neural Information Processing Systems*, volume 25, 2011.

Yu, Y. and Schuurmans, D. Rank/norm regularization with closed-form solutions: Application to subspace clustering. In *Conference on Uncertainty in Artificial Intelligence*, 2011.