Unfolding Latent Tree Structures using 4th Order Tensors

Mariya Ishteva MARIYA.ISHTEVA@VUB.AC.BE
ELEC, Vrije Universiteit Brussel, 1050 Brussels, Belgium
Haesun Park, Le Song {HPARK,LSONG}@CC.GATECH.EDU
College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA

Abstract

Discovering the structures of latent variable models whose conditional independence structures are trees is an important yet challenging learning task. Existing approaches for this task often require the unknown number of hidden states as an input. In this paper, we propose a quartet based approach which is *agnostic* to this number. The key contribution is a novel rank characterization of the tensor associated with the marginal distribution of a quartet. This characterization allows us to design a *nuclear norm* based test for resolving quartet relations. We then use the quartet test as a subroutine in a divide-and-conquer algorithm for recovering the latent tree structure. We also derive the conditions under which the algorithm is consistent and its error probability decays exponentially with increasing sample size. We demonstrate that the proposed approach compares favorably to alternatives. In a real world stock dataset, it also discovers meaningful groupings of variables, and produces a model that fits the data better.

1. Introduction

Discovering the latent structures from many observed variables is an important yet challenging learning task. The discovered structures can help better understand the domain and lead to potentially better predictive models. Many local search heuristics based on maximum parsimony and maximum likelihood methods have been proposed to address this problem (Semple & Steel, 2003; Zhang, 2004; Heller & Ghahramani, 2005; Teh et al., 2008; Harmeling & Williams, 2010). Their common drawback is the lack of consistency guarantees. Furthermore, the number of hidden states often needs to be determined in advance or estimated by time consuming cross-validations.

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

We focus on latent variables models whose conditional independence structures are trees (called latent tree models; an example for stock data is shown in Fig. 1). For these models, efficient algorithms with provable performance guarantees have been explored in the phylogenetic tree reconstruction community. One popular algorithm is the neighbor-joining (NJ) algorithm (Saitou & Nei, 1987), where pairs of variables are joined recursively according to a certain distance measure. The NJ algorithm is consistent when the distance measure satisfies the path additive property (Mihaescu et al., 2009). For discrete random variables, the additive distance is defined using the determinant of the joint probability table of a pair of variables (Lake, 1994). However, this definition only applies when the observed and the latent variables have the same number of states. If the latent variables represent simpler factors with smaller number of states, the NJ algorithm can perform poorly.

Another family of provably consistent methods is the quartet-based methods (Semple & Steel, 2003; Erdös et al., 1999). These methods first resolve a set of latent relations for quadruples of observed variables (quartets), and subsequently, stitch them together to form a latent tree. A good quartet test plays an essential role in these methods, as it is called repeatedly by the stitching algorithms. Recently, Anandkumar et al. (2011) proposed a quartet test using the leading k singular values of the joint probability table, where k is the number of hidden states. This approach allows k to be different from the number of observed states. However, it still requires k to be given in advance.

Our goal is to design a latent structure discovery algorithm which is *agnostic* to the number of hidden states, since in practice we rarely know this number. The proposed approach is quartet based, where the quartet relations are resolved based on rank properties of 4th order tensors associated with the joint

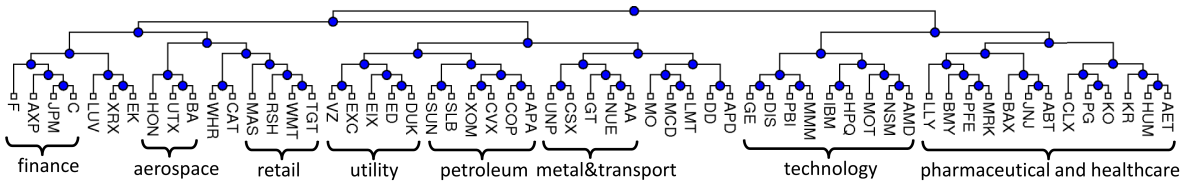


Figure 1. Latent tree structure estimated from stock data. The price of each stock is considered as a variable and these variables are connected via other latent variables in a graphical model with tree structure.

probability tables of quartets. The key insight is that rank properties of the tensor reveal the latent structure behind a quartet. Similar observations have been reported in the phylogenetic community (Eriksson, 2005; Allman & Rhodes, 2006), but they are concerned about the cases where the number of hidden states is larger or equal to the number of observed states. We focus instead on the cases where the number of hidden states is smaller, representing simpler factors. Furthermore, if the joint probability tensor is only approximately given (due to sampling noise) the main rank condition has to be modified. In Allman & Rhodes (2006) such condition is missing and in Eriksson (2005) the condition is heuristically translated to the distance of a matrix to its best rank-k approximation. In contrast, we propose a novel nuclear norm relaxation of the rank condition, discuss its advantages, and provide recovery conditions and finite sample guarantees. Our quartet test is easy to compute since it only involves singular value decomposition of unfolded 4th order tensors.

Using the proposed quartet test as a subroutine, the latent tree structure can be recovered in a divide-and-conquer fashion (Pearl & Tarsi, 1986). For d observed variables, the computational complexity of the algorithm is $O(d \log d)$, making it scalable to large problems. Under mild conditions, the tree construction algorithm using our quartet test is consistent and stable to estimate given a finite number of samples. In simulations, we compared to alternatives in terms of resolving quartet relations and building the entire latent trees. The proposed approach is among the best performing ones while being agnostic to the number of hidden states k. The latter is an important improvement, since cross validation for finding k is expensive while leading to similar final results. We also applied the new approach to a stock dataset, where it discovered meaningful grouping of stocks according to industrial sectors, and led a latent variable model that fits the data better than the competitors.

2. Latent Tree Graphical Models

We focus on discrete latent variable models whose conditional independence structures are trees. We assume all d observed variables, $\{X_1, \ldots, X_d\}$, are leaves of the tree, having the same number of states, n. We also assume all d_h hidden variables, $\{X_{d+1}, \ldots, X_{d+d_h}\}$, have the same, *but unknown*, number of states, k, ($k \leq n$).[1] We use uppercase letters for random variables (*e.g.*, X_i) and lowercase letters their instantiations (*e.g.*, x_i).

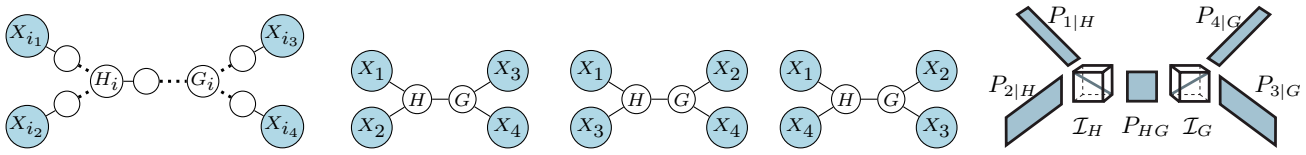
Factorization of distribution. The joint distribution of all $d + d_h$ variables in a latent tree model is a multi-way table (tensor) $\mathcal{P}$ of order $d + d_h$. Although the tensor has $O(n^d k^{d_h})$ entries, they can be computed from just a polynomial number of parameters due to the latent tree structure. That is, $\mathcal{P}(x_1, \ldots, x_{d+d_h}) = \prod_{i=1}^{d+d_h} P(x_i|x_{\pi_i})$, where each $P(X_i|X_{\pi_i})$ is a conditional probability table (CPT) of a variable X_i and its parent X_{π_i} in the tree.[2] This factorization leads to a significant saving in terms of representation parameters: we can represent exponential number of entries by just $O(d_h k^2 + dnk)$ parameters from the CPTs. Throughout the paper, we assume that **(A1)** all CPTs have full column rank, k, and the marginal distributions of all variables have full support. This assumption is needed for the identifiability of the latent variable models, and is common in latent tree recovery literature (Anandkumar et al. (2011)). Furthermore, it is needed only for the later rank conditions but not for the nuclear norm relaxation.

Structure learning. Determining the tree topology is an important and challenging learning problem. The goal is to discover the latent structure based just on samples of observed variables. For simplicity and uniqueness of the topology (Pearl, 1988), we assume that **(A2)** every latent variable has *exactly* 3 neighbors. This assumption can potentially be relaxed (§5).

Quartet. A quadruple of observed variables from a latent tree $\mathcal{T}$ is called a quartet (Fig. 2(a)). Given condition **(A2)**, there are 3 ways to connect a quartet, X_1, X_2, X_3, X_4, using 2 latent variables H and G (Fig. 2(b)). However, only one of the 3 quartet relations is consistent with $\mathcal{T}$. The mapping between quartets and the tree topology $\mathcal{T}$ is captured in the following theorem (Buneman, 1971):

[1] Our results can be generalized to the case where hidden variables have different states (see discussion in §A13). For simplicity of presentation, we assume their states are equal.

[2] In a latent tree, we can select a latent node as root, and reorient all edges away from it, inducing consistent parent-child relations. For the root X_r, $P(X_r|X_{\pi_r}) = P(X_r)$.



(a) Quartet (X_1, X_2, X_3, X_4) (b) $\{\{1,2\},\{3,4\}\}$ $\{\{1,3\},\{2,4\}\}$ $\{\{1,4\},\{2,3\}\}$ (c) Tensor $\mathcal{P}(X_1, X_2, X_3, X_4)$

Figure 2. (a) A quartet from a tree. (b) Three fixed ways to connect X_1, X_2, X_3 and X_4 with two latent variables. (c) Schematic diagram of the factorization of tensor $\mathcal{P}(X_1, X_2, X_3, X_4)$).

Theorem 1 *The set of all quartet relations $\mathcal{Q}_\mathcal{T}$ is unique to a latent tree $\mathcal{T}$, and furthermore, $\mathcal{T}$ can be recovered from $\mathcal{Q}_\mathcal{T}$ in polynomial time.*

Quartet-based tree reconstruction. Motivated by Theorem 1, a family of latent tree recovery algorithms has been designed based on resolving quartet relations. These algorithms first determine one of the 3 ways how 4 variables are connected, and then join together all quartet relations to form a consistent latent tree. For a model with d observed variables, there are $O(d^4)$ quartet relations in total (taking all possible combinations of 4 variables). However, we do not necessarily need to resolve all these quartet relations in order to reconstruct the latent tree. A small set of size $O(d \log d)$ will suffice for the tree recovery, which makes quartet based methods efficient even for problems with large d (Pearl & Tarsi, 1986; Pearl, 1988). In this paper, we design a new quartet based method. Our main contribution compared to previous approaches is that our method is *agnostic* to the number of hidden states, k, which is usually unknown in practice.

3. Resolving Quartet Relations without Having the Number of Hidden States

In this section, we develop a test for resolving the latent relation of a quartet when the number of hidden states is unknown. Our approach uses information from the joint probability table of a quartet, which is a 4th order tensor. Suppose the quartet relation of X_1, X_2, X_3 and X_4 is $\{\{1,2\},\{3,4\}\}$, then the tensor's entries are specified by $\mathcal{P}(x_1, x_2, x_3, x_4) =$

$$\sum\nolimits_{h,g} P(x_1|h)P(x_2|h)P(h,g)P(x_3|g)P(x_4|g). \quad (1)$$

This factorization suggests that there exist some low rank structures in the 4th order tensor. To study the rank properties of $\mathcal{P}(X_1, X_2, X_3, X_4)$, we first relate it to the conditional probability tables, $P(X_1|H)$, $P(X_2|H)$, $P(X_3|G)$, $P(X_4|G)$, and the joint probability table, $P(H, G)$ (we abbreviate them as $P_{1|H}$, $P_{2|H}$, $P_{3|G}$, $P_{4|G}$ and P_{HG}, respectively). Using tensor algebra, we have $\mathcal{P}(X_1, X_2, X_3, X_4) = \langle \mathcal{T}_1, \mathcal{T}_2 \rangle_3$,

$$\text{with} \quad \begin{array}{l} \mathcal{T}_1 = \mathcal{I}_H \times_1 P_{1|H} \times_2 P_{2|H}, \\ \mathcal{T}_2 = \mathcal{I}_G \times_1 P_{3|G} \times_2 P_{4|G} \times_3 P_{HG}, \end{array}$$

where $\mathcal{I}_H$ and $\mathcal{I}_G$ are 3rd order diagonal tensors of size $k \times k \times k$ with diagonal elements equal to 1. The multiplication $\times_i$ denotes a tensor-matrix multiplication with respect to the i-th dimension of the tensor and the rows of the matrix, and $\langle \cdot, \cdot \rangle_3$ denotes tensor-tensor multiplication along the third dimension of both tensors[3] (see illustration in Fig. 2(c)). Next we will characterize the rank properties of $\mathcal{P}$ and then exploit them to design a quartet test for latent structure discovery.

3.1. Unfolding the 4th Order Tensor

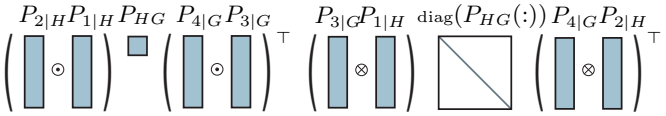
Now we consider 3 different reshapings A, B and C of the tensor into matrices ("unfoldings"). These unfoldings contain exactly the same entires as $\mathcal{P}$ but in different order. A corresponds to the grouping $\{\{1,2\},\{3,4\}\}$ of the variables, *i.e.*, the rows of A correspond to dimensions 1 and 2 of $\mathcal{P}$, and its columns to dimensions 3 and 4. B corresponds to the grouping $\{\{1,3\},\{2,4\}\}$ and C - to the grouping $\{\{1,4\},\{2,3\}\}$. Using MATLAB's notation (see §A8 for further explanation),

$$A = \text{reshape}(\mathcal{P}, n^2, n^2); \quad (2)$$
$$B = \text{reshape}(\text{permute}(\mathcal{P}, [1,3,2,4]), n^2, n^2); \quad (3)$$
$$C = \text{reshape}(\text{permute}(\mathcal{P}, [1,4,2,3]), n^2, n^2). \quad (4)$$

Next we present useful characterizations of A, B and C, essential for understanding their connection with the latent structure of a quartet. The *Kronecker product* of two matrices M and M' is denoted as $M \otimes M'$, and if they have the same number of columns, their *Khatri-Rao product* (column-wise Kronecker product), is denoted as $M \odot M'$. Then (see §A9 for proof),

Lemma 2 *Assume that $\{\{1,2\},\{3,4\}\}$ is the correct latent structure. The matrices A, B and C can be factorized respectively as (see Fig. 3 for illustration)*

$$A = \left(P_{2|H} \odot P_{1|H}\right) \; P_{HG} \; \left(P_{4|G} \odot P_{3|G}\right)^\top, \quad (5)$$
$$B = \left(P_{3|G} \otimes P_{1|H}\right) \text{diag}(P_{HG}(:)) \left(P_{4|G} \otimes P_{2|H}\right)^\top, \quad (6)$$
$$C = \left(P_{4|G} \otimes P_{1|H}\right) \text{diag}(P_{HG}(:)) \left(P_{3|G} \otimes P_{2|H}\right)^\top. \quad (7)$$



(a) A (b) B

Figure 3. Schematic diagrams of the unfoldings A and B.

The factorization of A is very different from those of B and C. First, in A, $P_{2|H} \odot P_{1|H}$ is a matrix of size

[3]For formal definitions of tensor notations see §A8.

$n^2 \times k$, and the columns of $P_{2|H}$ interact only with their corresponding columns in $P_{1|H}$. However, in B, $P_{3|G} \otimes P_{1|H}$ is a matrix of size $n^2 \times k^2$, and every column of $P_{1|H}$ interacts with every column of $P_{3|G}$ respectively (similarly for C). Second, in A, the middle factor P_{HG} has size $k \times k$, whereas in B, the entires of P_{HG} appear as the diagonal of a matrix of size $k^2 \times k^2$ (similarly for C). These differences result in different rank properties of A, B and C which we will exploit to discover the latent structure of a quartet.

3.2. Rank Properties of the Unfoldings

Given condition **(A1)** that all CPTs have full column rank, the factorization of A, B and C in (5), (6) and (7) respectively suggest that (see §A9 for more details)

$$\begin{aligned} &\text{rank}(A) = \text{rank}(P_{HG}) = k \\ &\leq \text{rank}(B) = \text{rank}(C) = \text{nnz}(P_{HG}), \end{aligned} \tag{8}$$

where $\text{nnz}(\cdot)$ denotes the number of nonzero elements. We note that the equality is attained if and only if the relationship between the hidden variables G and H is deterministic, *i.e.*, there is a single nonzero element in each row and in each column of P_{HG}. In this case, the grouping of variables in a quartet can be arbitrary, and we will not consider this case in the paper. More specifically, we have

Theorem 3 *Assume P_{HG} has a few zero entries, then $k \ll k^2 \approx \text{nnz}(P_{HG})$ and thus*

$$\boxed{\text{rank}(A) \ll \text{rank}(B) = \text{rank}(C).} \tag{9}$$

The above theorem reveals a useful difference between the correct grouping of variables and the two incorrect ones. Furthermore, this condition can be easily verified: Given $\mathcal{P}$ we can check the rank of its matrix representations A, B and C and thus discover the latent structure of the quartet.

3.3. Nuclear Norm Relaxation

In practice, due to sampling noise, all three matrices A, B and C would be full rank, so the rank condition cannot be applied directly. To deal with this, we relax the rank condition using nuclear norm $\|M\|_* = \sum_{i=1}^{n} \sigma_i(M)$, which is the sum of all singular values of an $(n \times n)$ matrix M. Instead of comparing the ranks of A, B and C, we look for the matrix with the smallest nuclear norm and declare the latent structure corresponding to it. This simple quartet algorithm is summarized in Algorithm 1.

Note that Algorithm 1 works even if the number of hidden states, k, is a priori unknown. This is an important advantage over the idea of learning the structure based on additive distance (Lake, 1994), where k is assumed to be the same as the number of states, n, of the observed variables, or over a recent approach based on

Algorithm 1 $i^* = \text{Quartet}(X_1, X_2, X_3, X_4)$

1: Estimate $\widehat{\mathcal{P}}(X_1, X_2, X_3, X_4)$ from a set of m *i.i.d.* samples $\{(x_1^l, x_2^l, x_3^l, x_4^l)\}_{l=1}^m$.
2: Unfold $\widehat{\mathcal{P}}$ in three different ways into matrices $\widehat{A}$, $\widehat{B}$ and $\widehat{C}$, and compute their nuclear norms $a_1 = \|\widehat{A}\|_*$, $a_2 = \|\widehat{B}\|_*$ and $a_3 = \|\widehat{C}\|_*$.
3: Return $i^* = \text{argmin}_{i \in \{1,2,3\}} a_i$.

quartet test (Anandkumar et al., 2011), where k needs to be specified in advance.

In our current context, nuclear norm has a few useful properties. First, it is the tightest convex lower bound of the rank of a matrix (Fazel et al., 2001). This is why[4] it is meaningful to compare nuclear norms instead of ranks. Second, it is easy to compute: a standard singular value decomposition will do the job. Third, it is robust to estimate. The nuclear norm of a probability matrix $\widehat{A}$ based on samples is nicely concentrated around its population quantity (Rosasco et al., 2010). Given a confidence level $1 - 2e^{-\tau}$, an estimate based on m samples satisfies

$$\begin{aligned} &|\|A\|_* - \|\widehat{A}\|_*| = \\ &\left|\sum\nolimits_i \sigma_i(A) - \sum\nolimits_i \sigma_i(\widehat{A})\right| \leq 2\sqrt{2\tau}/\sqrt{m}. \end{aligned} \tag{10}$$

Fourth, the nuclear norm can be viewed as a measure of dependence between two pairs of variables. For instance, A corresponds to grouping $\{\{1,2\},\{3,4\}\}$, and $\|A\|_*$ measures the dependence between the compound variables $\{X_1, X_2\}$ and $\{X_3, X_4\}$. In the community of kernel methods, A is treated as a cross-covariance operator between $\{X_1, X_2\}$ and $\{X_3, X_4\}$, and its spectrum has been used to design various dependence measures, such as Hilbert-Schmidt Independence Criterion, which is the sum of squares of all singular values (Gretton et al., 2005a), and kernel constrained covariance, which only takes the largest singular value (Gretton et al., 2005b). Intuitively, our quartet test says that: if we group the variables correctly, then cross group dependence should be low, since the groups are separated by two latent variables; however if we group the variables incorrectly, then cross group dependence should be high, since similar variables exist in the two groups.

4. Recovery Conditions and Finite Sample Guarantee for Quartets

Since nuclear norm is just a convex lower bound of the rank, there might be situations where the nuclear

[4]Note that A, B and C contain the same elements so their Frobenius norms are the same, *i.e.*, the 3 matrices are equally "normalized".

norm does not satisfy the same relation as the rank. That is, it might happen that $\text{rank}(A) \leq \text{rank}(B)$ but $\|A\|_* \geq \|B\|_*$. Next, we present sufficient conditions under which nuclear norm returns successful test.

When latent variables H and G are independent, $\text{rank}(P_{HG}) = 1$, since $P_{HG} = P_H P_G^\top$ ($P(h,g) = P(h)P(g)$). Let $\{\{1,2\},\{3,4\}\}$ be the correct quartet relation. We can obtain simpler characterizations of the 3 unfoldings of $\mathcal{P}(X_1, X_2, X_3, X_4)$, denoted as $A_\perp$, $B_\perp$ and $C_\perp$ respectively. Using Lemma 2 and the independence of H and G (see appendix, (27)–(28)),

$$\begin{aligned} A_\perp &= (P_{2|H} \odot P_{1|H})\ P_H P_G^\top\ (P_{4|G} \odot P_{3|G})^\top \\ &= P_{12}(:)\ P_{34}(:)^\top, \end{aligned} \tag{11}$$

$$\begin{aligned} B_\perp &= (P_{3|G} \otimes P_{1|H})(\text{diag}(P_G) \otimes \text{diag}(P_H))(P_{4|G} \otimes P_{2|H})^\top \\ &= P_{34} \otimes P_{12}, \end{aligned} \tag{12}$$

and $\text{rank}(A_\perp) = 1 \ll \text{rank}(B_\perp)$ which is consistent with Theorem 3. Furthermore, since $A_\perp$ has only one nonzero singular value, we have $\|A_\perp\|_* = \|A_\perp\|_F = \|B_\perp\|_F \leq \|B_\perp\|_*$ (using $\|M\|_F \leq \|M\|_*$ for any matrix M). Similarly, $C_\perp = P_{43} \otimes P_{12}$ and $\|A_\perp\|_* \leq \|C_\perp\|_*$. Then we know for sure that the nuclear norm quartet test will return the correct topology.

When latent variables H and G are not independent, we treat it as perturbation Δ away from the independent case, *i.e.*, $\widetilde{P}_{HG} = P_H P_G^\top + \Delta$. The size of Δ quantifies the strength of dependence between H and G. Obviously, when Δ is small, *e.g.*, $\Delta = \mathbf{0}$, we are back to the independence case and it is easy to discover the correct quartet relation; when it is large, *e.g.*, $\Delta = I - P_H P_G^\top$, H and G are deterministically related and the different groupings are indistinguishable. The question is how large can Δ be while still allowing the nuclear norm quartet test to find the correct latent relation.

First, from the definition of Δ, we have $\Delta\mathbf{1} = \mathbf{0}$, and $\Delta^\top \mathbf{1} = \mathbf{0}$, where $\mathbf{1}$ and $\mathbf{0}$ are vectors of all ones and all zeros. Thus, the perturbation Δ does not affect the marginal distributions P_H and P_G, since $\widetilde{P}_H = \widetilde{P}_{HG}\mathbf{1} = P_H P_G^\top \mathbf{1} + \Delta\mathbf{1} = P_H$. Assuming $\{\{1,2\},\{3,4\}\}$ is the correct quartet relation, Δ does not affect the pairwise marginal distribution P_{12} neither, since $P_{12} = P_{1|H}\,\text{diag}(P_H) P_{2|H}^\top$ and the marginal P_H is the same before and after the perturbation. Similar reasoning also applies to $P_{34} = P_{3|G}\,\text{diag}(P_G) P_{4|G}^\top$.

We define *excessive dependence* of the correct and incorrect groupings as

$$\theta := \min\{\|B_\perp\|_* - \|A_\perp\|_*,\ \|C_\perp\|_* - \|A_\perp\|_*\}.$$

It quantifies the changes in dependence when we switch from incorrect groupings to the correct one (in the case when H and G are independent). Note that θ is measured only from pairwise marginals (11)-(12), P_{12} and P_{34}. Using matrix perturbation analysis we can show that (see appendix §11 for proof)

Lemma 4 *If $\|\Delta\|_F \leq \frac{\theta}{k^2+k}$, the minimum of $\|A\|_*$, $\|B\|_*$ and $\|C\|_*$ will reveal the correct quartet relation.*

Thus, if the excessive dependence θ is large compared to the number of hidden states, the size of the allowable perturbation can be correspondingly larger. In other words, if the dependence between variables within the same group is strong enough compared to the dependence across groups, we allow for larger Δ and stronger dependence between hidden variables H and G (which is closer to the indistinguishable case). It is difficult to directly compare our recovery conditions with previous work, since we are addressing the more difficult case where latent state k is unknown. Our recovery condition constrains the correlation between hidden variables based on observable quantity θ and the number of latent states k, while those of Anandkumar et al. (2011) assume the unobserved correlation ρ between latent variables is given.

Last, under the recovery condition in Lemma 4, and given m *i.i.d.* observations, we can obtain the following guarantee for the quartet test (see appendix, §14 for proof). Let $\alpha = \min\{\|B\|_* - \|A\|_*, \|C\|_* - \|A\|_*\}$.

Lemma 5 *With probability $1 - 8e^{-\frac{1}{32}m\alpha^2}$, Algorithm 1 returns the correct quartet relation.*

5. Building Latent Tree from Quartets

We can use the resolved quartet relations (Algorithm 1) to discover the structure of the entire tree via an incremental divide-and-conquer algorithm (Pearl & Tarsi, 1986; Pearl, 1988), summarized in Algorithm 2 (further details in appendix §10). Joining variable X_{i+1} to the current tree of i leaves can be done with $O(\log i)$ tests. This amounts to performing $O(d \log d)$ quartet tests for building an entire tree of d leaves, which is efficient even if d is large. Moreover, this algorithm is consistent (Pearl & Tarsi, 1986).

Tree recovery conditions and guarantees. When a quartet is taken from a latent tree, each edge of the quartet corresponds to a path in the tree involving a chain of variables (Fig. 2(a)). We need to bound the perturbation to each single edge of the tree such that joint path perturbations satisfy edge perturbation conditions from Lemma 4. For a quartet $q = \{\{i_1, i_2\},\{i_3, i_4\}\}$ corresponding to a single edge between H and G, denote the excessive dependence by θ_q. By adding perturbation Δ_q of size smaller than $\frac{\theta_q}{k^2+k}$ to $P_H P_G^\top$ we can still correctly

Algorithm 2 $\mathcal{T} = \text{BuildTree}(X_1, \ldots, X_d)$

1: Connect any 4 variables X_1, X_2, X_3, X_4 with 2 latent variables in a tree $\mathcal{T}$ using Algorithm 1.
2: **for** $i = 4, 5, \ldots, d-1$ **do** {insert $(i+1)$-th leaf X_{i+1}}
3: Choose root R that splits $\mathcal{T}$ into sub-trees $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ of roughly equal size.
4: Choose any triplet $(X_{i_1}, X_{i_2}, X_{i_3})$ of leaves from different sub-trees.
5: Test which sub-tree should X_{i+1} be joined to: $i^* \leftarrow \text{Quartet}(X_{i+1}, X_{i_1}, X_{i_2}, X_{i_3})$.
6: Repeat recursively from step 3 with $\mathcal{T} := \mathcal{T}_{i^*}$. This will eventually reduce to a tree with a single leaf. Join X_{i+1} to it via hidden variable.
7: **end for**

recover q. Let $\theta_{\min} := \min_{\text{quartet } q} \theta_q$. If we require $\|\Delta_q\|_F \leq \frac{\theta_{\min}}{k^2+k}$, all such quartet relations will be recovered successfully. If we further restrict the size of the perturbation by the smallest value in a marginal probability distribution of a hidden variable, $\gamma_{\min} := \min_{\text{hidden node } H} \min_{i=1\ldots k} P_H(i)$, we can guarantee that all quartet relations corresponding to a path between H and G can also be successfully recovered by the nuclear norm test (see appendix §12). The intuitive interpretation of γ_{min} is that if a hidden state rarely occurs (small probability), samples for the observed variables contain very little information about the hidden variable. It becomes harder to identify the latent structure in this case and hence smaller perturbation away from independence is allowed. Therefore, we assume that **(A3)** $\|\Delta_q\|_F \leq \min\{\frac{\theta_{\min}}{k^2+k}, \gamma_{\min}\}$ for all quartets q in a tree.

Theorem 6 *Given condition* **(A1)**–**(A3)** *and population quantities, algorithm 2 returns the correct tree topology.*

The recovery conditions guarantee that all quartet relations can be resolved correctly and simultaneously. Then a consistent algorithm using a subset of the quartet relations should return the correct tree structure. We note that condition **(A2)** could be relaxed to allow hidden variables to have more than 3 neighbors. In this case, instead of using the minimum of the the nuclear norm of A, B and C for quartet tests, we may need to consider their differences, *e.g.*, $\|A\|_* - \|B\|_*$, to decide whether to join the observed variables with one or with two variables, as in Anandkumar et al. (2011). This is left as our future work. Last, given m *i.i.d.* samples, we have the following statistical guarantee for the Algorithm 2 (see appendix, §15 for proof). Let $\alpha_{\min} := \min_{\text{quartet } q} \alpha_q$, and a constant c,

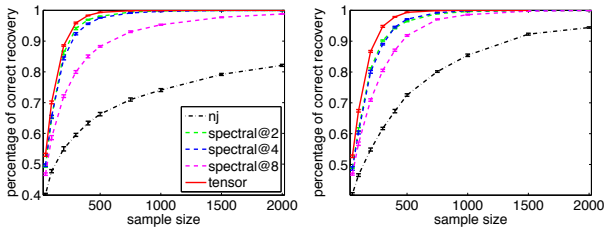
Theorem 7 *Given condition* **(A1)**–**(A3)** *and m samples, Algorithm 2 recovers the correct tree topology, with probability* $1 - 8 \cdot c \cdot d \log d \cdot e^{-\frac{1}{32} m \alpha_{\min}^2}$.

We note that the conditions needed for our results are stronger than those used by previous work (Anandkumar et al. (2011)). This is partly due to the fact that our method deals with a more difficult case where we do not know the number of hidden states. Another reason is that our analysis relies on the simple reconstruction algorithm by Pearl & Tarsi (1986). There are better quartet based algorithms for building latent trees with stronger statistical guarantees, *e.g.* Erdös et al. (1999). We can adapt our nuclear norm based quartet test to those algorithm as well. However, this is not the main focus of the paper. We choose the divide-and-conquer algorithm due to its simplicity, ease of analysis and it illustrates well how our quartet recovery guarantee can be translated into a guarantee for latent tree reconstruction.

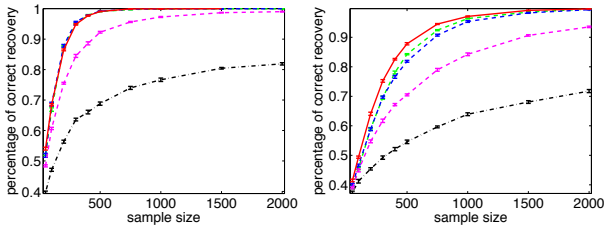
6. Experiments

We compared our algorithm with the neighbor-joining algorithm (NJ) (Saitou & Nei, 1987), a quartet based algorithm of Anandkumar et al. (2011) (Spectral@k), the Chow-Liu neighbor Joining algorithm (CLNJ) (Choi et al., 2011), and an algorithm of Harmeling & Williams (2010) (HW).

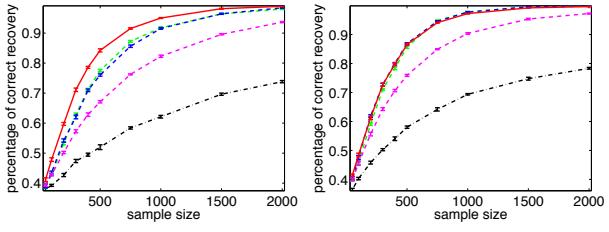
NJ proceeds by recursively joining two variables that are closest according to an additive distance defined as $d_{ij} = \frac{1}{2} \log\det \text{diag}\, P_i - \log|\det P_{ij}| + \frac{1}{2} \log\det \text{diag}\, P_j$, where "det" denotes determinant, "diag" is a diagonalization operator, P_{ij} denotes the joint probability table $P(X_i, X_j)$, and P_i and P_j the probability vector $P(X_i)$ and $P(X_j)$ respectively (Lake, 1994). When P_{ij} has rank $k < n$, $\log|\det P_{ij}|$ is not defined, NJ can perform poorly. Spectral@k uses singular values of P_{ij} to design a quartet test (Anandkumar et al., 2011). For instance, if the true quartet configuration is $\{\{1,2\},\{3,4\}\}$ as in Fig. 2(b), then the quartet needs to satisfy $\prod_{s=1}^k \sigma_s(P_{12})\sigma_s(P_{34}) > \max\{\prod_{s=1}^k \sigma_s(P_{13})\sigma_s(P_{24}), \prod_{s=1}^k \sigma_s(P_{14})\sigma_s(P_{23})\}$. Based on this relation, a confidence interval based quartet test is designed and used as a subroutine for a tree reconstruction algorithm. Spectral@k can handle cases with $k < n$, but still requires k as an input. We will show in later experiments that its performance is sensitive to the choice of k. CLNJ first applies Chow-Liu algorithm (Chow & Liu, 1968) to obtain a fully observed tree and then proceeds by adding latent variables using neighbor joining algorithm. The HW algorithm is a greedy algorithm to learn binary trees by iteratively joining two nodes with a high mutual information. The number of hidden states is automatically determined in the HW algorithm and can be different for different latent variables.



(a) $k = \{2, 4\}, \mu = 0.5$ (b) $k = \{2, 8\}, \mu = 0.5$

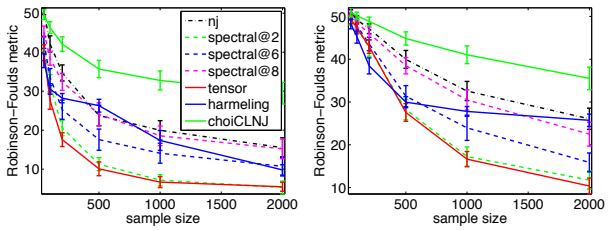


(c) $k = \{4, 4\}, \mu = 0.5$ (d) $k = \{2, 4\}, \mu = 1$

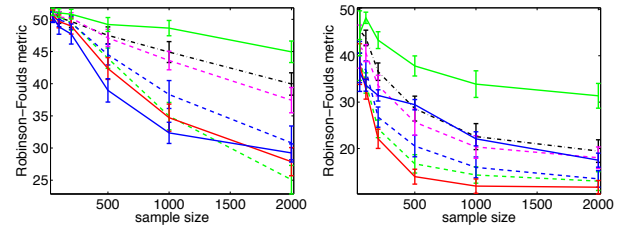


(e) $k = \{2, 8\}, \mu = 1$ (f) $k = \{4, 4\}, \mu = 1$

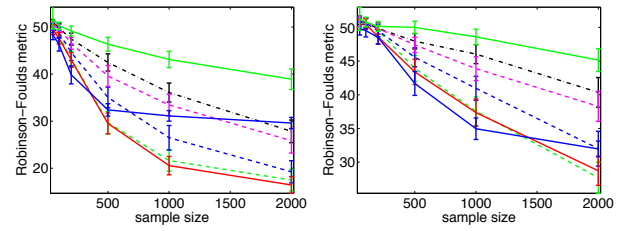
Figure 4. (a)-(f) Quartet recovery results.



(a) $\mu = 0.2, \beta = 0.5$ (b) $\mu = 0.5, \beta = 0.5$



(c) $\mu = 1, \beta = 0.5$ (d) $\mu = 0.2, \beta = 0.2$



(e) $\mu = 0.5, \beta = 0.2$ (f) $\mu = 1, \beta = 0.2$

Figure 5. (a)-(f) Tree recovery results.

6.1. Resolving Quartet Relations

We compared our method to NJ and Spectral@k in terms of their ability to recover the quartet relation among four variables. We used quartet with three different configurations for the hidden states: (1) $k_H = 2$, $k_G = 4$ (small difference); (2) $k_H = 2$, $k_G = 8$ (large difference); and (3) $k_H = k_G = 4$ (no difference). In all cases, the number of observed states was fixed to $n = 10$. We always started from P_{HG} corresponding to $H \perp G$, but $P_{X_i|H}$ corresponding to deterministically related X_i and H (similarly for $P_{X_i|G}$), and perturbed them using the following formula $P(a = i|b) = \frac{P(a=i|b)+u_i}{\sum_i P(a=i|b)+u_i}$, where all u_i are *i.i.d.* random variables drawn from Uniform$[0, \mu]$. We then drew random sample from the quartet according to these CPTs. We studied the percentage of correctly recovered quartet relations as we varied the sample size across $S = \{50, 100, 200, 300, 400, 500, 750, 1000, 1500, 2000\}$ and under two different levels of perturbation ($\mu = \{0.5, 1\}$). We randomly initialized each experiment 1000 times and report the average quartet recovery performance and the standard error in Fig. 4.

The proposed method compares favorably to NJ and Spectral@k. The performance of Spectral@k varies a lot depending on the chosen number of singular values k. Our method is free from tuning parameters and often stays among the top performing ones. Especially when the number of hidden states are very different from each other ($k_H = 2$ and $k_G = 8$), our method is leading the second best by a large gap (Fig. 4(b) and 4(e)). When both hidden states are the same ($k_H = k_G = 4$), the Spectral@k achieves the best performance when the chosen number of singular values k is the same as k_H. Note that allowing Spectral@k to use different k resembles using cross validations for finding the best k. It is expensive while our approach performs almost indistinguishable from Spectral@k even it choose the best k.

6.2. Discovering Latent Tree Structure

We used different tree topologies and sample sizes in this experiment. We generated tree topologies by randomly splitting 16 observed variables recursively into two groups. The recursive splitting stops when there are only two nodes left in a group. We introduced a hidden variable to join the two partitions in each recursion and this gives a latent tree structure. The topology of the tree is controlled by a single splitting parameter β which controls the relative size of the first partition versus the second. If β is close to 0 or 1, we obtain trees of skewed shape, with long path of hidden variables. If β is close to 0.5, the resulting latent trees are more balanced. In our experiments, we experimented with skewed latent trees $\beta = 0.2$ and balanced

Table 1. Negative log-likelihood ($\times 10^5$) on test data.

	Tensor	Spectral@k	Choi (CLNJ)	Neighbor-joining	Harmeling	Chow-Liu
$k=2$	4.4101 ± 0.0014	4.4415 ± 0.0035	4.4299 ± 0.0009	4.4258 ± 0.0015	4.3134 ± 0.0028	4.4067 ± 0.0028
$k=4$	4.3041 ± 0.0017	4.3464 ± 0.0038	4.3345 ± 0.0020	4.3294 ± 0.0026		
$k=6$	4.2835 ± 0.0017	4.3473 ± 0.0041	4.3162 ± 0.0016	4.3111 ± 0.0022		
$k=8$	$\mathbf{4.2829 \pm 0.0019}$	4.3526 ± 0.0037	4.3162 ± 0.0018	4.3104 ± 0.0028		
$k=10$	4.2854 ± 0.0022	4.3706 ± 0.0023	4.3185 ± 0.0018	4.3111 ± 0.0023		

trees $\beta = 0.5$. We first generate different random k between 2 and 8 for the hidden states, and then generate the probability models for each tree using the same scheme as in our previous experiment. Here we experimented with perturbation level $\mu = \{0.2, 0.5, 1\}$.

We varied the sample size across $S = \{50, 100, 200, 500, 1000, 2000\}$, and measured the error of the constructed tree using Robinson-Foulds metric (Robinson & Foulds, 1981). This measure is a metric over trees of the same number of leaves. It is defined as $(a + b)$ where a is the number of partitions of variables implied by the learned tree but not by the true tree and b is the number of partitions of the variables implied by the true tree but not by the learned tree (similar spirit to precision and recall).

The tree recovery results are shown in Fig. 5(a)-5(f). Again we can see that our proposed method compares favorably to existing algorithms. Throughout the 6 experimental conditions, the tensor approach and spectral@2 performed the best with sufficiently large sample sizes. Note that we tried out different k for Spectral@k which resembles using cross validations for finding the best k. Even in this case, our approach works comparably without having to know k. Harmeling-William's algorithm performed well in small sample sizes, while CLNJ does not perform well in these experimental conditions.

6.3. Understanding Latent Relations of Stocks

We applied our algorithm to discover a latent tree structure from a stock dataset. Our goal is to understand how stock prices X_i are related to each other. We acquired closing prices of 59 stocks from 1984 to 2011 (from www.finance.yahoo.com), which provides us 6800 samples. The daily change of each stock price is discretized into 10 values, and we applied our algorithm to build a latent tree. A visualization of the learned tree topologies and discovered groupings are shown in Fig. 1.

We see nice groupings of stocks according to their industrial sectors. For instance, companies related to petroleum, such as CVX (Chevron), XOM (Exxon Mobil), APA (Apache), COP (ConocoPhillips), SLB (Schlumberger) and SUN (Sunoco), are grouped into a subtree. Pharmaceutical companies, such as MRK (Merck), PFE (Pfizer), BMY (Bristol Myers Squibb), LLY (Eli Lilly), ABT (Abbott Laboratories), JNJ (Johnson and Johnson) and BAX (Baxter International), are all grouped into a subtree.

We also compared different algorithms in terms of held-out likelihood. We first randomized the data 10 times, and each time used half for training and half for computing the held-out likelihood. Then we estimated the latent binary tree structures using different algorithms. Finally, we fit latent variable models to the discovered structures. The number of the states for all hidden variables, k, were the same in each latent variable model. We experimented with $k = 2, 4, 6, 8, 10$ to simulate the process of using cross validation to select the best k. The results are presented in Table 1. Note that Harmeling-William's algorithm automatically discovers k, so it does not use the experimental parameter k. The Chow-Liu tree does not contain any hidden variables and hence just one number in the table. CLNJ and Neighbor-joining assume the states for the hidden and observed variables are the same during structure learning. However, in parameter fitting, we can still use different number of hidden states k. In this experiment, the structure produced by our tensor approach produced the best held-out likelihood.

7. Conclusion

We propose a quartet-based method for discovering the tree structures of latent variable models. The practical advantage of the new method is that we do not need to pre-specify the number of the hidden states, a quantity usually unknown in practice. The key idea is to view the joint probability tables of quadruple of variables as 4th order tensors and then use the spectral properties of the unfolded tensors to design a quartet test. We provide conditions under which the algorithm is consistent and its error probability decays exponentially with increasing the sample size. In both simulated and a real dataset, we demonstrated the usefulness of our methods for discovering latent structures. While in this study we focus on the properties of the 4th order tensor and its various unfoldings, we believe that properties of tensors and methods and algorithms from multilinear algebra will allow to address many other problems arising from latent variable models.

Acknowledgments: Research supported by ERC Grant 258581; Belgian Network DYSCO - IAP VII; US government; NSF IIS1218749; Georgia Tech Startup Fund.

References

Allman, E. S. and Rhodes, J. A. The identifiability of tree topology for phylogenetic models, including covarion and mixture models. *Journal of Computational Biology*, 13(5):1101–1113, 2006.

Anandkumar, A., Chaudhuri, K., Hsu, D., Kakade, S., Song, L., and Zhang, T. Spectral methods for learning multivariate latent tree structure. In *Neural Information Processing Systems*, 2011.

Buneman, P. The recovery of trees from measures of dissimilarity. In Hodson, F.R., Kendall, D.G., and Tautu, P. (eds.), *Mathematics in the archaeological and historical sciences*, pp. 387–395. Edinburgh University Press, 1971.

Carroll, J. and Chang, J. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

Choi, M., Tan, V., Anandkumar, A., and Willsky, A. Learning latent tree graphical models. *Journal of Machine Learning Research*, 12:1771–1812, 2011.

Chow, C., and Liu, C. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

Erdös, P. L., Székely, L. A., Steel, M. A., and Warnow., T. J. A few logs suffice to build (almost) all trees: Part II. *Theoretical Computer Science*, 221:77–118, 1999.

Eriksson, N. Tree construction using singular value decomposition. In Pachter, L. and Sturmfels, B. (eds.), *Algebraic Statistics for Computational Biology*, pp. 347–358. Cambridge University Press, 2005. URL http://dx.doi.org/10.1017/CBO9780511610684.

Fazel, Maryam, Hindi, Haitham, and Boyd, Stephen P. A rank minimization heuristic with application to minimum order system approximation. In *American Control Conference*, pp. 4734–4739, 2001.

Grasedyck, L. Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.*, 31(4): 2029–2054, 2010.

Gretton, A., Bousquet, O., Smola, A. J., and Schölkopf, B. Measuring statistical dependence with Hilbert-Schmidt norms. In Jain, S., Simon, H. U., and Tomita, E. (eds.), *Proceedings of the International Conference on Algorithmic Learning Theory*, pp. 63–77. Springer-Verlag, 2005a.

Gretton, A., Herbrich, R., Smola, A. J., Bousquet, O., and Schölkopf, B. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005b.

Harmeling, S. and Williams, C. Greedy learning of binary latent trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1087–1097, 2010.

Harshman, R. A. Foundations of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16(1):1–84, 1970.

Heller, K. A. and Ghahramani, Z. Bayesian hierarchical clustering. In *Proceedings of the International Conference on Machine Learning*, pp. 297–304, 2005.

Lake, J.A. Reconstructing evolutionary trees from dna and protein sequences: paralinear distances. *Proceedings of the National Academy of Sciences*, 91 (4):1455, 1994.

Mihaescu, R., Levy, D., and Pachter, L. Why neighbor-joining works. *Algorithmica*, 54(1):1–24, 2009.

Oseledets, I. V. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33:2295–2317, 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.

Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, 1988.

Pearl, J. and Tarsi, M. Structuring causal trees. *Journal of Complexity*, 2(1):60–77, 1986.

Robinson, D.F. and Foulds, L.R. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2): 131–147, 1981.

Rosasco, L., Belkin, M., and Vito, E.D. On learning with integral operators. *Journal of Machine Learning Research*, 11:905–934, 2010.

Saitou, N. and Nei, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987.

Semple, C. and Steel, M.A. *Phylogenetics*, volume 24. Oxford University Press, USA, 2003.

Teh, Yee Whye, Daume, Hal, and Roy, Daniel. Bayesian agglomerative clustering with coalescents. In *Advances in Neural Information Processing Systems 22*, 2008.

Zhang, N. L. Hierarchical latent class models for cluster analysis. *Journal of Machine Learning Research*, 5:697–723, 2004.