An Algorithm to Learn Polytree Networks with Hidden Nodes

Firoozeh Sepehr†, Donatello Materassi†

†University of Tennessee, Knoxville

Motivation

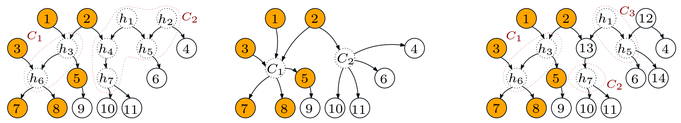
Ancestral graphs are a prevalent mathematical tool to take into account latent (hidden) variables in a probabilistic graphical model. In ancestral graph representations, the nodes are only the observed (manifest) variables and the notion of m-separation fully characterizes the conditional independence relations among such variables, bypassing the need to explicitly consider latent variables. However, ancestral graph models do not necessarily represent the actual causal structure of the model, and do not contain information about, for example, the precise number and location of the hidden variables. Being able to detect the presence of latent variables while also inferring their precise location within the actual causal structure model is a more challenging task that provides more information about the actual causal relationships among all the model variables, including the latent ones. Here, we develop an algorithm to exactly recover graphical models of random variables with underlying polytree structures when the latent nodes satisfy specific degree conditions. Therefore, this article proposes an approach for the full identification of hidden variables in a polytree.

Preliminaries and Assumptions

A latent polytree $\vec{P}_\ell = (V, L, \vec{E})$ is minimal if every hidden node $y_h \in L$ satisfies one of the following conditions:

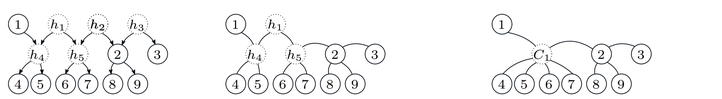
- $\deg^+_{\vec{P}_\ell}(y_h) \geq 2$ and $\deg_{\vec{P}_\ell}(y_h) \geq 3$ and if $|\mathrm{pa}_{\vec{P}_\ell}(y_h)| = 1$, then $\mathrm{pa}_{\vec{P}_\ell}(y_h) \subseteq V$;
- $\deg^+_{\vec{P}_\ell}(y_h) = 2$ and $\deg^-_{\vec{P}_\ell}(y_h) = 0$ and $\deg^-_{\vec{P}_\ell}(y_{c_1}), \deg^-_{\vec{P}_\ell}(y_{c_2}) \geq 2$ where $\mathrm{ch}_{\vec{P}_\ell}(y_h) = \{y_{c_1}, y_{c_2}\}$.

We define the hidden clusters in a hidden polytree as subsets of hidden nodes that are connected to each other only through hidden nodes. Also, when the hidden clusters are replaced by a single hidden node, we call the obtained structure the collapsed representation of the latent polytree.



(A Latent Polytree) (The Collapsed Polytree) (A Minimal Latent Polytree)

We also define two types of hidden nodes in order to be able to recover all the hidden nodes in a minimal latent polytree properly. Furthermore, the quasi-skeleton of a latent polytree is obtained when the orientations as well as the type-II hidden nodes are removed.



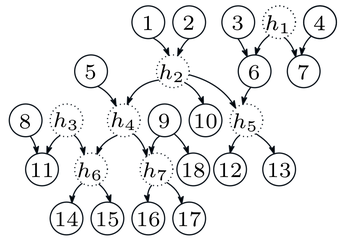
(A Latent Polytree $\vec{P}_\ell$) (Quasi-skeleton of $\vec{P}_\ell$) (Collapsed Quasi-skeleton of $\vec{P}_\ell$)

Problem Statement

Assume a semi-graphoid defined over a set of variables $V \cup L$. Let the latent polytree $\vec{P}_\ell = (V, L, \vec{E})$ be faithful to the semi-graphoid and assume that the nodes in L satisfy the minimality conditions. Recover the pattern of $\vec{P}_\ell$ from conditional independence relations involving only nodes in V.

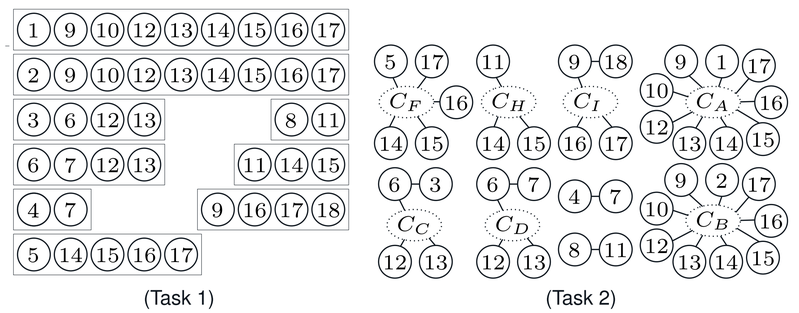
Proposed Algorithm

Consider the latent polytree demonstrated in the following figure.



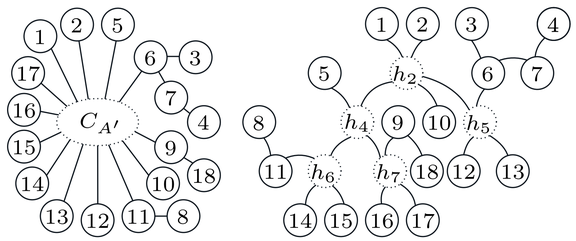
(A Latent Polytree)

Our algorithm for learning the pattern of a minimal latent polytree is made of 5 tasks.
Task 1: Using the independence statements involving only the visible nodes, determine the number of rooted subtrees in the latent polytree and their respective sets of visible nodes;
Task 2: Given all the visible nodes belonging to each rooted subtree, determine the collapsed quasi-skeleton of each rooted subtree;



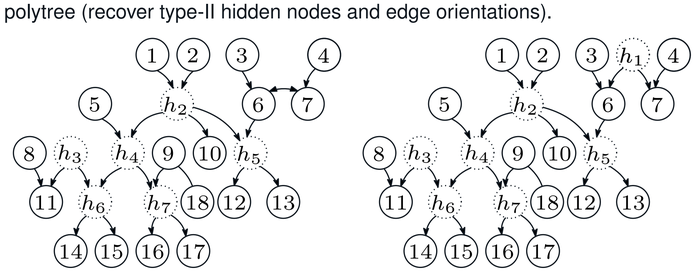
(Task 1) (Task 2)

Task 3: Merge the overlapping hidden clusters in the collapsed quasi-skeleton of each rooted subtree to obtain the collapsed quasi-skeleton of the latent polytree;
Task 4: Determine the quasi-skeleton of the latent polytree from the collapsed quasi-skeleton of the latent polytree (recover type-I hidden nodes);



(Task 3) (Task 4)

Task 5: Obtain the pattern of the latent polytree from the recovered quasi-skeleton of the latent polytree (recover type-II hidden nodes and edge orientations).



(Task 5a) (Task 5b)

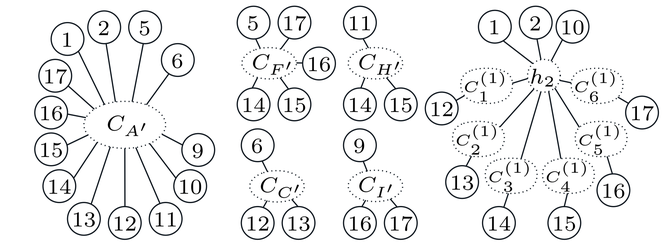
More Details

In order to implement Task 4, we need to define two tests for the identification of the following nodes:

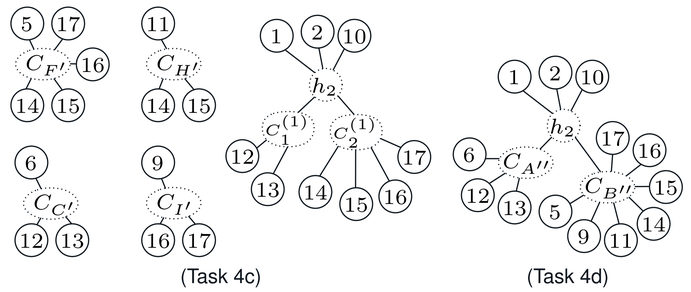
- Rooted subtree T_r contains a hidden root of the hidden cluster C if and only if $\tilde{V}_r \neq \emptyset$ and for all $\tilde{V}_{r'}$ with $r' \neq r$ we have $|\tilde{V}_r \setminus \tilde{V}_{r'}| > 1$ or $|\tilde{V}_{r'} \setminus \tilde{V}_r| \leq 1$.
- The visible nodes linked to y_h are given by the set $W \setminus \overline{W}$ where

$$I := \{r\} \cup \{r' \text{such that } |\tilde{V}_r \setminus \tilde{V}_{r'}| = |\tilde{V}_{r'} \setminus \tilde{V}_r| = 1\},$$
$$W := \bigcup_{i \in I} \tilde{V}_i, \quad \overline{W} := \bigcup_{i \notin I} \tilde{V}_i.$$

Using these two tests, we are able to recursively apply the same approach to recover the structure of the hidden clusters in Task 4. These details are shown in the following figures.



(Task 4a) (Task 4b)



(Task 4c) (Task 4d)

Conclusions

We have provided an algorithm to reconstruct the pattern of a latent polytree graphical model. The algorithm only requires the second and third order statistics of the observed variables and no prior information about the number and location of the hidden nodes is assumed. An important property of the proposed approach is that the algorithm is sound under specific degree conditions on the hidden variables. If such degree conditions are not met, it is shown that there exists another latent polytree with fewer number of hidden nodes entailing the same independence relations.

Acknowledgements

This work has been partially supported by NSF (CNS CAREER #1553504).