# Learning Bayesian networks with unobserved variables

Pekka Parviainen, Mehmood Alam Khan

April 2, 2013

## 1 Background

Bayesian networks are often used to analyze causal relations between variables. While using Bayesian networks in causal analysis one often assumes causal sufficiency, that is, that no unobserved variable affects two or more observed variables. However, this is often an unrealistic assumption. Therefore, applying standard models straightforwardly can lead to misleading conclusions. Thus, there is demand for models that handle unobserved (hidden, latent) variables.

There are two main approaches to structure learning in Bayesian networks: constraint-based and score-based. The constraint-based model is based on testing conditional independencies between variables and constructing networks that express the found independencies (and dependencies). On the other hand, the score-base approach is based on assigning each network a score based on how well the network fits to the data and trying to find a highest-scoring network.

While score-based methods perform often very well in practice, especially when the data is scarce, they lack a principled approach for handling unobserved variables. The constraint-based approach, on the other hand, allows presenting the structures using maximal ancestral graphs (MAGs) that implicitly include unobserved variables.

Models with unobserved variables are challenging due to the fact that there are an infinite amount of such models. Thus, to be able use such models in practice, we need to make some assumptions. However, finding a "good" set of assumptions is challenging.

In mathematical modeling we often have to balance between model complexity and computational complexity. Generally, the more complex the model is the better it is able to catch all the relevant aspects of the phenomenon that we try to model. However, complex models can easily become computationally intractable. Thus, the goal is to construct models that are complex enough to capture all relevant aspects of the phenomenon of interest while being simple enough to allow inference in practice.

### 1.1 Prerequisites

This project requires understanding of several key concepts in Bayesian networks and Bayesian probability theory. At least the following concepts should be understood.

Bayesian networks: d-separation, Markov equivalence, identifiability, faithfulness, perfect map [**?**, ch. 3], structure learning [**?**, ch. 18], partially observed data [**?**, ch. 19.4, 19.5], causality [**?**, ch. 21].

Bayesian probability theory: marginalization, uncertainty represented by a probability distribution, Bayesian approach to Bayesian networks [**?**].

There have been some studies on Bayesian networks with unobserved variables [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**].

## 2 Research questions

The goal of this project is to construct a principled and practical Bayesian model for Bayesian networks with unobserved variables. Basically, the project has three "dimensions": theoretical, algorithmic and practical. The main research questions can be summarized as follows.

1. What is the expressiveness of the model (compared to other models)?
2. How to learn the model effectively?
3. How good is the model in practice?

The first two questions are highly interdependent.

## 3 Solution ideas

To get started, we can consider simplified models. For example, we can try to solve the problem when the structure for observed variables forms a (poly)tree. We can also allow at most one hidden variable per observed variable.

Another possibility is to consider some information-theoretic framework instead of the Bayesian one.

## 4 Problem Formulation

**Notations:**

- '$x_i$' denotes observed variable and 'X' as set of observed variables.
- 'D' denotes dataset. $D = \{x_i[k] \mid i \in [1, ..., n], k \in [1, ..., m]\}$
- '$y_{\langle i,j \rangle}$' denotes unobserbed variable and 'Y' as set of unobserved variables. $H = \{y_{\langle i,j \rangle}[k] \mid \{i, j\} \in (X \times X), k \in [1, ..., m]\}$

- $G = (V, A)$ is a directed acyclic graph with vertex set V and set of arcs A. Where $V = [X \cup Y]$
- The parent set of a variable is denoted by $p$. For-instance, the parent set of variable $x_i$ in a graph G will be $p_G(x_i)$
- Similarly, Children of a variable $x_i$ in a graph G is represented by $c_G(x_i)$
- $\Theta_{x_i}$ are the parameters for variable $x_i$ of X in a given Bayesian network.
- $\Theta_{x_i} = \{\Theta_{x_i|p_G(x_i)}\}$ are the parameters for variable $x_i$ in a BN given its parent set.
- $\Theta_{x_i|p_G(x_i)} = \{\theta_{x_i|p_G(x_i)=\mathbf{u}}\}$ are parameters for variable $x_i$ given its parent set take their **u**-th configuration in a given bayesian network.
- Similarly, $\Theta_G = \{\Theta_{x_i}\}, i = 1, .., n$, encodes all the parameters for a given bayesian network with underlying graph G.

**Assumptions:**

- A hidden variable $y_{\langle i,j \rangle}$ can not have any parents but it may have zero or two children, $x_i$ and $x_j$.
- An observed variable can have many parents or none.
- Complete data.
- *Multinomial sample:* Dataset D consists of multinomial samples. Binary variables are special cases. Same is assumed for hidden variables H.
- *Dirichlet:* Prior over $\Theta_{x_i|p_G(x_i)}$ is a *Dirichlet* distribution.
- *Parameter independence:* The priors over the parameters $\{\theta_{x_i|p_G(x_i)=\mathbf{u}}\}$ for different $x_i, p_G$ and $u$ are independent.
  - *Global parameter independence:* Given a graph G such that $P(G) > 0$ then, $P(\Theta_G|G) = \prod_i^n P(\Theta_{x_i}|G)$
  - *Local parameter independence:* Given a graph G such that $P(G) > 0$ then, $P(\Theta_{x_i}|G) = \prod_{u \in U} P\left(\theta_{x_i|p_G(x_i)=u}|G\right)$,
    where $U = \{u_i\}$ are the set of all possible configuration of parents of $x_i$
- *Parameter modularity:* For a given two DAGs, $G$ and $G^{'}$, such that $P(G) > 0$ and $P\left(G^{'}\right) > 0$, If a variable $x_i$ has the same parent set in $G$ and $G^{'}$ i.e. $p_G(x_i) = p_{G'}(x_i) = \mathbf{U}$, then
  $P\left(\theta_{x_i|U}|G\right) = P\left(\theta_{x_i|U}|G^{'}\right)$

**Structure learning with fully observed data:**

$$P\left(G|D\right)=\int_{\theta}\frac{P\left(D|G,\theta\right)P\left(\theta|G\right)P\left(G\right)}{P\left(D\right)}\,\mathrm{d}\theta \tag{1}$$

where, G refers to a structure or DAG, D as data and $\theta$ are parameters.

$$P\left(G|D\right)\propto\int_{\theta}P\left(D|G,\theta\right)P\left(\theta|G\right)P\left(G\right)\,\mathrm{d}\theta \tag{2}$$

**Structure learning with partially observed data:**

$$P\left(G|D\right)=\sum_{H}\int_{\theta}\frac{P\left(G\right)P\left(\theta|G\right)P\left(H|\theta,G\right)P\left(D|H,\theta,G\right)}{P\left(D\right)}\,\mathrm{d}\theta \tag{3}$$

where, H is hidden variable and assumed as discrete.

$$P\left(G|D\right)\propto\sum_{H}\int_{\theta}P\left(G\right)P\left(\theta|G\right)P\left(H|\theta,G\right)P\left(D|H,\theta,G\right)\,\mathrm{d}\theta \tag{4}$$

**Components:**

1. $P(G)$**:**

$$P\left(G\right)\propto\prod_{i\in N}s\left(x_{i},p_{G}\left(x_{i}\right)\right)\prod_{\{i,j\}\in(N\times N)}r\left(y_{\langle i,j\rangle},c_{G}\left(y_{\langle i,j\rangle}\right)\right) \tag{5}$$

Where,

$$r\left(y_{\langle i,j\rangle},c_{G}\left(y_{\langle i,j\rangle}\right)\right)=\begin{cases}\geq 0 & \text{if } c_{G}\left(y_{\langle i,j\rangle}\right)=\emptyset \text{ OR } c_{G}\left(y_{\langle i,j\rangle}\right)=\{i,j\}\\ 0 & \text{Otherwise}\end{cases} \tag{6}$$

$$s\left(x_{i},p_{G}\left(x_{i}\right)\right)=\begin{cases}0 & \text{if } \{y_{\langle i,j\rangle}\in p_{G}\left(i\right),j\neq i,k\neq i\}\\ & \text{OR}\\ & \{y_{\langle i,j\rangle}\in p_{G}\left(i\right)\text{ AND }x_{j}\in p_{G}\left(i\right)\}\\ & \text{OR}\\ & \{y_{\langle j,i\rangle}\in p_{G}\left(i\right)\text{ AND }x_{j}\in p_{G}\left(i\right)\}\\ \geq 0 & \text{Otherwise}\end{cases} \tag{7}$$

2. $P(\Theta|G)$**:**

$$P(\Theta|G)=\prod_{i}^{n}f(\theta_{x_{i}|p_{G}(x_{i})}|G)\prod_{\langle i,j\rangle\in(N\times N)}\rho\left(\theta_{\{i,j\}}\right) \tag{8}$$

3. $P(H|\Theta, G)$:

$$P(H|\Theta, G) = \prod_{k=1}^{m} \prod_{\langle i,j\rangle \in (N\times N)} \theta_{y_{\langle i,j\rangle [k]}} \tag{9}$$

4. $P(D|H, \Theta, G)$:

$$P(D|H, \Theta, G) = \prod_{k=1}^{m} \prod_{i}^{n} \theta_{x_i|p_G(x_i)=u[k]} \tag{10}$$

# 5 Full Joint Distribution

$$P(D, H, \Theta, G) = P(G)P(\Theta|G)P(H|\Theta|G)P(D|H, \Theta|G) \tag{11}$$

$$\begin{aligned} = \prod_{i\in N} s\left(x_i, p_G\left(x_i\right)\right) \prod_{\{i,j\}\in (N\times N)} r\left(y_{\langle i,j\rangle}, c_G\left(y_{\langle i,j\rangle}\right)\right) \prod_{i}^{n} f(\theta_{x_i|p_G(x_i)}|G) \\ \prod_{\langle i,j\rangle \in (N\times N)} \rho\left(\theta_{\{i,j\}}\right) \prod_{k=1}^{m} \prod_{\langle i,j\rangle \in (N\times N)} \theta_{y_{\langle i,j\rangle [k]}} \prod_{k=1}^{m} \prod_{i}^{n} \theta_{x_i|p_G(x_i)=u[k]} \end{aligned} \tag{12}$$

$$\begin{aligned} = \prod_{i\in N} \left( s\left(x_i, p_G\left(x_i\right)\right) f(\theta_{x_i|p_G(x_i)}|G) \prod_{k=1}^{m} \theta_{x_i|p_G(x_i)=u[k]} \right) \\ \prod_{\{i,j\}\in (N\times N)} \left( r\left(y_{\langle i,j\rangle}, c_G\left(y_{\langle i,j\rangle}\right)\right) \rho\left(\theta_{\{i,j\}}\right) \prod_{k=1}^{m} \theta_{y_{\langle i,j\rangle [k]}} \right) \end{aligned} \tag{13}$$

## 5.1 Algorithm

1. Gready equivalance search for observed variables $X$
2. Repeat until no improvement
   - For every member $A'$ in equivalance class of $A$
     - For $e \in A'$, compute score of the network given that $e$ is replaced with hidden parent. Where $e$ is an arc whose both ends are observed. Replacement approach can be optimal or sum over all the values.
   - Choose the best modification and assign it to $A$.
3. Repeat step 2 but instead removing hidden variables one by one.