208891 Probabilistic Graphical Models **Spring 2019**

Lecture 9: February 3

Lecturer: Donlapark Pornnopparath

9.1 Bayesian score of Bayesian networks

Let G be an arbitrary Bayesian network. For each variable X_i, $k \in \text{Val}(X_i)$ and its observed set of parents $\boldsymbol{u}_i$, let $\alpha_{k|\boldsymbol{u}_i}$ be the corresponding hyperparameter of the Dirichlet prior. Define

$$\alpha_{X_i|\boldsymbol{u}_i} = \sum_{k \in \text{Val}(X_i)} \alpha_{k|\boldsymbol{u}_i}.$$

Then the like likelihood function given data D is

$$P(D|G) = \prod_i \prod_{\boldsymbol{u}_i \in \text{Val}(\text{Pa}(X_i))} \frac{\Gamma(\alpha_{X_i|\boldsymbol{u}_i})}{\Gamma(\alpha_{X_i|\boldsymbol{u}_i} + n(\boldsymbol{u}_i))} \prod_{k \in \text{Val}(X_i)} \left[\frac{\Gamma(\alpha_{k|\boldsymbol{u}_i} + n(k, \boldsymbol{u}_i))}{\Gamma(\alpha_{k|\boldsymbol{u}_i})} \right]$$

By using the Stirling's approximation $\log \Gamma(x) \approx \frac{1}{2}\log(2\pi)x \log x - \frac{1}{2}\log x - x$, we obtain

$$\log P(D|G) = LL(\hat{\boldsymbol{\theta}}_G|D) - \frac{\log n}{2}|G| + O(1)$$

as $M \to \infty$. Here, $|G|$ is the number of independent parameters in G. This motivates the *BIC score*:

$$s_{BIC} = LL(\hat{\boldsymbol{\theta}}_G|D) - \frac{\log n}{2}|G|,$$

which can be rewritten, using the decomposition of the log-likelihood function,

$$s_{BIC} = n \sum_{i=1}^{n} I_{\widehat{P}}(X_i, \text{Pa}(X_i)) - n \sum_{i=1}^{n} H_{\widehat{P}}(X_i) - \frac{\log n}{2}|G|.$$

Hence, the BIC score grows almost linearly with the number of examples. The first term measures the fit of G to the data , while the last term penalizes the model complexity. For a large value of n, the model emphasize more on the fit to the data than the complexity.

9.2 Priors

9.2.1 Structure priors

Now we study the prior over network structures $P(G)$. As the number of examples grows, the prior will be relatively small compared to the marginal likelihood. Consequently, we usual assume the uniform prior when there are large samples.

Alternatively, we could impose a prior that penalizes more complex graphs by defining

$$P(G) \propto c^{|E(G)|},$$

where $0 < c < 1$ and $|E(G)|$ is the number of edges in G.

For small samples, we might try the prior that has the form of

$$P(G) \propto \prod_i P\left(\boldsymbol{U}_i = \mathrm{Pa}(X_i)\right).$$

where $P\left(\boldsymbol{U}_i = \mathrm{Pa}(X_i)\right)$ is the prior probability that we assign to the set of parents $\mathrm{Pa}(X_i)$ of X_i. This prior allows us to consider the local graphs separately without disturbing the global properties (such as the depth of G).

9.2.2 Parameter priors

Lastly, we have to specify the hyperparameter $\alpha_{x_i|\boldsymbol{u}_i}$ of the parameter priors. The most basic approach is to take the Dirichlet distribution $\mathrm{Dir}(\alpha, \alpha, \ldots, \alpha)$ as parameters priors, where the number of α's is equal to the size of the sample space. For example, with a simple binary node Y we might consider $\mathrm{Dir}(1, 1)$ in which case we assume that the imaginary sample size is two. However, if we consider an addition of a parent X of Y which has four possible values. Then $\mathrm{Dir}(1, 1)$ would assume eight different imaginary samples, two for each parameter $\theta_{Y|x^i}$. This result does not match the reality as the number of imaginary samples should not increase with the addition of X.

A better approach is to specify an imaginary sample size α and a prior distribution P' that follows our belief or previous studies. If we have no prior knowledge, then P' is usually set to be uniform. Then, we set the parameters as follows:

$$\alpha_{x_i|\boldsymbol{u}_i} = \alpha P'(x_i, \boldsymbol{u}_i)$$

Then, the number of imaginary samples that satisfy $Y = y$ can be computed as follows:

$$\alpha_y = \alpha P'(y) = \sum_{x_i} \alpha P'(y, x_i) = \sum_{x_i} \alpha_{y|x_i},$$

and so the sum is independent of the choices of parents for Y.

9.3 Structure search

In this section, we assume that a training set D, a scoring function s and a set of all possible network structures $\mathcal{G}$ have already been obtained. Additionally, we assume that the score function can be analyze at the local level:

Definition 9.1. A score function s is *decomposable* if for any $G \in \mathcal{G}$,

$$s(G|D) = \sum_i \mathrm{Score}(X_i|\mathrm{Pa}(X_i), D).$$

Note that the score s_L and s_B with priors and marginal likelihood discussed in previous sections are all decomposable.

Our goal is to find a graph $G \in \mathcal{G}$ that maximizes $s(G|D)$.

9.3.1 Learning a forest

We consider a simple case where the network is a forest.

Definition 9.2. A Bayesian network G is a forest if for each variable X in G, $|\text{Pa}(X)| \leq 1$.

Instead of maximizing $s(G|D)$ directly, we consider the difference of a tree G and an empty graph G_0.

$$\begin{aligned}\Delta G &= s(G|D) - s(G_0|D) \\ &= \sum_i \text{Score}(X_i|\text{Pa}(X_i), D) - \text{Score}(X_i|D)\end{aligned}$$

These difference vanishes if $\text{Pa}(X_i) = \emptyset$, otherwise there is a parent X_j of X_i, leading to

$$\Delta G = \sum_{X_j \to X_i} \text{Score}(X_i|X_j, D) - \text{Score}(X_i|D).$$

Denoting $\text{Score}(X_i|X_j, D) - \text{Score}(X_i|D) = w_{j\to i}$, we can treat $w_{j\to i}$ as a weight on the graph. Our goal is not to find a forest in G that maximizes

$$\Delta G = \sum_{X_j \to X_i} w_{j\to i}.$$

This is a *maximum weight spanning forest* problem which can be solved with $O(m^2 \log m)$ where m is the number of variables.

Example 9.3. The red edges in Figure 9.1 cover the maximum weight spanning forest in the network with the total sum of weights 31. ◇

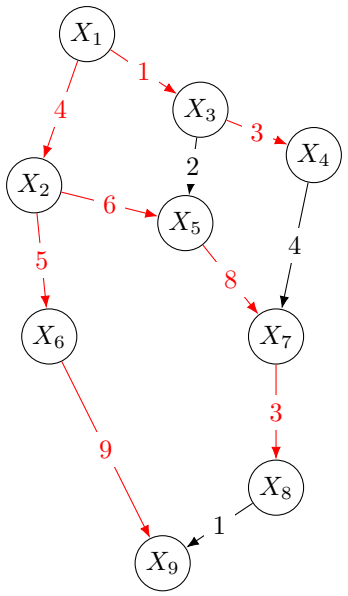

Figure 9.1: The subgraph spanned by the red edges is the maximum weight spanning forest.