Bayesian network tasks:

(a) Learning: learn the structure of a BN, \( B = \{ G, \Theta \} \)

Recall: \( G \) - acyclic graph, \( \Theta \) - parameters (conditional probab.)

(b) Inference: given the model \( B \), infer unobserved \( X_j \)

Learning Bayesian networks

Given a training set \( D = \{ x^1, x^2, \ldots, x^m \} \) of independent instances of \( X \), find a network \( B = \{ G, \Theta \} \) that best matches \( D \)

Occasionally, some of the values in vectors \( x^i \) may be missing

\( \Rightarrow \) incomplete data

Learning scenarios:

(a) Complete data (all instances are known)

(1) Known structure: statistical parametric estimation

(2) Unknown structure: discrete optimization

(b) Incomplete data

(1) Known structure: parametric optimization (EM, gradient search)

(2) Unknown structure: combined (mixture models, structural EM)

Learning structure \( G \)

Bayesian score:

\[
S(G; D) = \log P(G; D) = \log P(D | G) + \log P(G) + C,
\]

\( C \) - constant, \( P(D | G) = \int P(D | G, \Theta) P(\Theta | G) d\Theta \)

\( S(G; D) \) is asymptotically consistent: given enough data, true structure will receive a higher score than other graphs
Priors should be:
(a) structure equivalent, i.e., \( G \Leftrightarrow G' \Rightarrow P(G) = P(G') \)
(b) decomposable

If decomposable:
\[ S(G; D) = \sum_i \text{Score Contribution} (X_i, \text{Pa}(X_i); D) \]

Finding optimal \( G \) is NP-hard; heuristic techniques:
(a) simulated annealing
(b) greedy hill-climbing - \( O(m^2) \), decomposability helps

Quantifying confidence in the estimated \( G \): bootstrap method

Generate "perturbed" versions of the original data

The procedure:

1. Set \( M \), the \# of perturbed versions of data

2. For \( m = 1: M \),
   2.1 resample the data instances to obtain \( D_m \)
   2.2 apply the learning procedure on \( D_m \) to obtain \( \hat{G}_m \)

3. For each feature \( f \) (i.e., edge and its direction), compute

\[
\text{conf}(f) = \frac{1}{M} \sum_{m=1}^{M} f(\hat{G}_m),
\]

where
\[
f(\hat{G}_m) = \begin{cases} 
1, & \text{if } f \in \hat{G}_m \\
\emptyset, & \text{otherwise}
\end{cases}
\]
Learning parameters $\Theta$

Approaches: ML estimation, Bayesian learning

Example: A 4-node BN, 5 observations

$$D = \begin{bmatrix}
X(1) & X_1(1) & \ldots & X_n(1) \\
\vdots & \vdots & & \vdots \\
X(M) & X_1(M) & \ldots & X_n(M)
\end{bmatrix}$$

$$= \begin{bmatrix}
\vdots \\
\end{bmatrix}$$

ML estimation approach: assuming i.i.d. samples,

$$L(\Theta; D) \sim \prod_m P(E(m), B(m), A(m), C(m); \Theta)$$

$$= \prod_m P(E(m)) P(B(m)) P(A(m)|E(m), B(m)) P(C(m)|A(m))$$

$$= \prod_m P(E(m)) \prod_m P(B(m)) \prod_m P(A(m)|E(m), B(m)) \prod_m P(C(m)|A(m))$$

Started: the product of prob. of all nodes, for all m

Finished: the product of the likelihoods of each node

Generalized decomposition principle:

$$L(\Theta; D) = \prod_m P(x_1(m), \ldots, x_n(m)) = \prod_i \prod_m P(x_i(m) | Pa_i(m)) = \prod_i L_i(\Theta_i; D)$$

So, we compute the likelihoods separately for each node and then combine them

Note:

$$L(\Theta_i; D) = \prod_m P(x_i(m) | Pa_i(m)) = \prod_{Pa_i} \prod_m P(x_i(m) | Pa_i(m))$$

$$= \prod_{Pa_i} \prod_{x_i} P(x_i | Pa_i) \prod_{x_{\bar{i}}} P(x_{\bar{i}} | Pa_{\bar{i}})$$

$$= \prod_{Pa_i} \prod_{x_i} P(x_i | Pa_i) \prod_{Pa_{\bar{i}}} P(x_{\bar{i}} | Pa_{\bar{i}})$$
ML estimate:
\[ \hat{\delta}_{x_i | p_a} = \frac{N(x_i, p_i)}{N(p_a)} \]

\( N(x_i, p_a) \) - the # of occurrences of \( x_i \) given parent config. \( p_a \);
\( N(p_a) = \sum_{x_i} N(x_i, p_a) \)

Bayesian learning: consider all possible \( \Theta \) and their probabilities
\[ P(\Theta | x(n), ..., x(M)) = \frac{P(x(n), ..., x(M) | \Theta) P(\Theta)}{\int P(x(n), ..., x(M) | \Theta) P(\Theta) d\Theta} \tag{*} \]

Assume Dirichlet's priors:
\[ P(\Theta) = \frac{(\prod_{j=1}^{k} \lambda_j^{\alpha_j - 1})!}{(\lambda_1 - 1)! (\lambda_2 - 1)! \cdots (\lambda_k - 1)!} \prod_{j=1}^{k} \frac{\lambda_j^{\alpha_j - 1}}{\alpha_j} \propto \prod_{j=1}^{k} \theta_j^{\alpha_j - 1} \]
\( \lambda_1, \lambda_2, ..., \lambda_k \) - hyperparameters (presumed known)

Recall:
\[ L(\Theta; D) \sim P(D | \Theta) = \prod_{j=1}^{k} \theta_j^{N_j} \]

Now,
\[ P(\Theta | D) \propto P(\Theta) P(D | \Theta) \propto \prod_{j=1}^{k} \theta_j^{N_j + \alpha_j - 1} \]
\[ \propto \prod_{j=1}^{k} \theta_j^{\alpha_j - 1} \]

So, the posterior is Dirichlet as well.

Finding conditional expectation of \( \Theta \) given \( D \):
\[ \hat{\delta}_{x_i | p_a} = \frac{\lambda(x_i, p_a) + N(x_i, p_a)}{\lambda(p_a) + N(p_a)} \]
\( \lambda(x_i, p_a) = \lambda(x_i = j, p_a) \) - a coeff. in the Dirichlet distr. of \( \delta_i \)
\( \lambda(p_a) = \sum_{l} \lambda(x_i = l, p_a) \)