COMP538: Introduction to Bayesian Networks

Lecture 8: Structure Learning

Nevin L. Zhang
lzhang@cse.ust.hk

Department of Computer Science and Engineering
Hong Kong University of Science and Technology

Fall 2008

Objectives

- Discuss how to learn Bayesian network structures.
- Problem statement:
  - Given:
    - A set of random variables X_1, X_2, ..., X_n.
    - A data set on those variables.
  - Find: A Bayesian network (structure + parameters) that is "optimal" or "good" in some sense.
- Reading: Zhang and Guo (2007), Chapter 8.
- Reference: Geiger *et al.* (1996), Chickering and Heckerman (1997), Lanternman (2001), Friedman (1997)

Outline

The Problem of Model Selection

- Notations:
  - S — a candidate BN structure,
  - θ_S — vector of parameters for S.
- A BN structure encapsulates assumptions about how variables are related.Hence sometimes called a **model**.
- **Model selection problem** :
  - Given data **D**, what structure S should we choose?

Motivating the Principle

- Maximum likelihood principle for parameter estimation:
  - Choose parameters to maximize the loglikelihood $l(\theta|\mathbf{D}) = logP(\mathbf{D}|\theta)$.

- Loglikelihood of (S, θ_S) given data $\mathbf{D}$:

$$l(S, \theta_S|\mathbf{D}) = logP(\mathbf{D}|S, \theta_S)$$

- Choose structure and parameters to maximize the loglikelihood: Find (S^*, θ_S^*) such that

$$l(S^*, \theta_S^*|\mathbf{D}) = \sup_{S,\theta_S} l(S, \theta_S|\mathbf{D}) = \max_S \sup_{\theta_S} l(S, \theta_S|\mathbf{D})$$

Motivating the Principle

- Given S, we know how to find θ_S^* that maximizes $l(S, \theta_S|\mathbf{D})$. (MLE of parameters)
- The **maximized loglikelihood** of S given $\mathbf{D}$ is

$$l^*(S|D) = \sup_{\theta_S} l(S, \theta_S|\mathbf{D}) = l(S, \theta_S^*|\mathbf{D})$$

- Model selection: Choose structure (model) to maximize the maximized loglikelihood.
- Note: The word "maximize" applies to structure while the word "maximized" applies to parameters.

Property of Maximized likelihood

- Assume complete data.
- What structure would maximize the maximized likelihood?
- From Lecture 6, we know

$$l(S, \theta_S|\mathbf{D}) = \sum_{i,k} \sum_{j} m_{ijk} log \theta_{ijk},$$

where m_{ijk} is the number of data cases where $X_i = j$ and $pa_S(X_i) = k$.

- We also know that

$$\theta^*_{ijk} = \frac{m_{ijk}}{\sum_j m_{ijk}}$$

- Hence

$$l^*(S|\mathbf{D}) = l(S, \theta^*_S|\mathbf{D}) = \sum_{i,k} \sum_{j} m_{ijk} log \frac{m_{ijk}}{\sum_j m_{ijk}}$$

Property of Maximized likelihood

- Let $\hat{P}(\mathbf{X})$ be the **empirical distribution**:

$$\hat{P}(\mathbf{X}{=}\mathbf{x}) = \text{ fraction of cases in } \mathbf{D} \text{ where } \mathbf{X}{=}\mathbf{x}$$

- Let N be the sample size.

$$\hat{P}(X_i{=}j, pa_S(X_i) = k) = \frac{m_{ijk}}{N}$$

So

$$m_{ijk} = N\hat{P}(X_i{=}j, pa_S(X_i) = k)$$

$$\frac{m_{ijk}}{\sum_j m_{ijk}} = \hat{P}(X_i{=}j|pa_S(X_i) = k)$$

Property of Maximized likelihood

- Hence

$$
\begin{aligned}
I^*(S|\mathbf{D}) &= \sum_{i,k}\sum_{j} m_{ijk} log \frac{m_{ijk}}{\sum_j m_{ijk}} \\
&= \sum_{i}\sum_{j,k} N\hat{P}(X_i{=}j, pa_S(X_i) = k) log \hat{P}(X_i{=}j|pa_S(X_i) = k) \\
&= -N\sum_{i}\sum_{j,k} \hat{P}(X_i{=}j, pa_S(X_i) = k) log \frac{1}{\hat{P}(X_i{=}j|pa_S(X_i) = k)} \\
&= -N\sum_{i} H_{\hat{P}}(X_i|pa_S(X_i))
\end{aligned}
$$

Property of Maximized likelihood

- Let S' be the same as S except that certain X_i has one more parent, say, Y.
- From Theorem 1.5, we know that

$$H_{\hat{P}}(X_i|pa_{S'}(X_i)) = H_{\hat{P}}(X_i|pa_S(X_i), Y) \leq H_{\hat{P}}(X_i|pa_S(X_i))$$

  where the equality holds iff $X_i \perp_{\hat{P}} Y|pa_S(X_i)$.
- Because of randomness in the empirical distribution,$X_i \perp_{\hat{P}} Y|pa_S(X_i)$ is false with probability 1.
- Hence with probability 1:

$$l^*(S'|\mathbf{D}) > l^*(S|\mathbf{D})$$

Property of Maximized likelihood

- In general, more complex a model is, the better the maximized score.
- Maximized likelihood leads to over-fitting.
  - Under this criterion, the best model is the complete BN where each node is the parent of all its non-parents.

Outline

Learning trees

- A Bayesian network is **tree structured** if each variable has no more than one parent.
- For simplicity, call such Bayesian nets **trees**.
- Don't confuse trees with **polytrees**

  DAGs whose underlying undirected graphs contain no loop.

Learning trees

- **V**: a set of variables.
- **D**: a collection of complete data cases on the variables.
- Let $\mathcal{T}$ be the set of all possible trees of the variables.
- Consider the following problem:

  Find a tree $T^* \in \mathcal{T}$ that maximizes the maximized loglikelihood score, i.e.

$$l^*(T^*|\mathbf{D}) = \max_{T \in \mathcal{T}} l^*(T|\mathbf{D})$$

- Notes:
  - Overfitting is not a problem here because we restrict to $\mathcal{T}$.
  - Used quite often.

Learning trees

- We have already learned that

$$l^*(T|\mathbf{D}) = -N \sum_i H_{\hat{P}}(X_i|pa_T(X_i))$$

where N is the sample size and $\hat{P}$ is the empirical distribution based on $\mathbf{D}$.

- Using basic facts of Information Theory (Lecture 1), we have

$$\begin{aligned} l^*(T|\mathbf{D}) &= -N \sum_{i,pa_T(X_i)\neq\emptyset} (H_{\hat{P}}(X_i) - I_{\hat{P}}(X_i : pa_T(X_i)) - N \sum_{i,pa_T(X_i)=\emptyset} H_{\hat{P}}(X_i) \\ &= N \sum_{i,pa_T(X_i)\neq\emptyset} I_{\hat{P}}(X_i : pa_T(X_i)) - N \sum_i H_{\hat{P}}(X_i) \end{aligned}$$

- Let $G = (\mathbf{X}, E)$ be the undirected graph underlying T. Then

$$l^*(T|\mathbf{D}) = N \sum_{(X,Y)\in E} I_{\hat{P}}(X : Y) - N \sum_{X\in\mathbf{X}} H_{\hat{P}}(X)$$

Learning trees

- Trees with the same underlying undirected graphs have the same maximized loglikelihood score.
  They are hence **equivalent** and we cannot distinguish between them based on data.
- Our task becomes:
  - Find the undirected tree $G = (\mathbf{X}, E)$ that maximizes

$$I^*(G|\mathbf{D}) =_{\text{def}} N \sum_{(X,Y)\in E} I_{\hat{P}}(X:Y) - N \sum_{X\in\mathbf{X}} H_{\hat{P}}(X)$$

  - Note:

    $I_{\hat{P}}(X:Y)$ *is almost never zero. Hence, the optimal tree is connected tree.*

Learning trees

- Note the second term in $l^*(G|\mathbf{D})$ does not depend on the graph. So our task is really to find an undirected graph G to maximize:

$$N \sum_{(X,Y)\in E} I_{\hat{P}}(X:Y)$$

- This equivalent to the task of find the maximum spanning tree for the following weighted and undirected graph over $\mathbf{X}$:
  - There is an edge between each pair X and Y of variables in $\mathbf{X}$.
  - The weight on the edge is $I_{\hat{P}}(X:Y)$.
- There are two commonly used algorithms to find maximum spanning trees (Rosen, K. H. (1995). *Discrete Mathematics and Its Applications*. McGraw-Hill, Inc., New York, NY, third edition, 1995.)

Learning trees

- Kruskal's Algorithm.
  - Start with the empty graph and add edges one by one.
  - As the next edge to add, choose one that
    - Is not in graph yet.
    - Does not introduce a cycle.
    - Has the maximum weight.
- Prim's algorithm
  - Start with a graph containing one node and add edges and vertices one by one.
  - To figure out what to add next,
    - Go through edges that involve one vertex already in graph and one not in graph.
    - Add the edges (and hence a vertex) with the maximum weight.

Learning trees

- The materials described above are credited to

  Chow, C. K. and Liu, C. N. (1968). Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory, *IT-14(3), 462-467.*

- So, the results are called Chow-Liu trees.

Outline

The Principle

- View S and θ_S as random variables.
- Assume prior $P(S, \theta_S)$. This is the same as
  - Assume **structural prior**: $P(S)$, and
  - Assume **parameter prior**: $P(\theta_S|S)$

$$P(S, \theta_S) = P(\theta_S|S)P(S)$$

- Compute posterior:

$$P(S, \theta_S|\mathbf{D}) \propto P(\mathbf{D}|S, \theta_S)P(\theta_S|S)P(S)$$

Model Averaging

- Predicting the next case D_{m+1}:

$$\begin{aligned} P(D_{m+1}|\mathbf{D}) &= \sum_S \int P(D_{m+1}|S,\theta_S)P(S,\theta_S|\mathbf{D})d\theta_S \\ &= \sum_S \int P(D_{m+1}|S,\theta_S)P(S|\mathbf{D})P(\theta_S|S,\mathbf{D})d\theta_S \\ &= \sum_S P(S|\mathbf{D}) \int P(D_{m+1}|S,\theta_S)P(\theta_S|\mathbf{D},S)d\theta_S \qquad (1) \end{aligned}$$

- Note that we know how to compute the following from Bayesian parameter estimation:

$$\int P(D_{m+1}|S,\theta_S)P(\theta_S|\mathbf{D},S)d\theta_S$$

- Equation (1) averages predictions by different models. The operation hence called **model averaging**.
- Many possible models. Average over only top, say, 10 models.

Bayesian Score

- Model averaging typically is computationally difficult.
- So, prediction usually is based only on one model, the best model,
  - The one that maximizes $P(S|\mathbf{D})$.
- Note that

$$P(S|\mathbf{D}) = \frac{P(\mathbf{D}, S)}{P(\mathbf{D})} = \frac{P(\mathbf{D}|S)P(S)}{P(\mathbf{D})}$$

- $P(D)$ does not help with model selection. So we can select models using:

$$logP(\mathbf{D}, S) = logP(\mathbf{D}|S) + logP(S)$$

- This is the **Bayesian score** of S.

Marginal Likelihood

In the Bayesian score,

$$logP(\mathbf{D}, S) = logP(\mathbf{D}|S) + logP(S)$$

- $P(S)$ is the structural prior.
- And

$$\begin{aligned} P(\mathbf{D}|S) &= \int P(\mathbf{D}|S, \theta_S) P(\theta_S|S) d\theta_S \\ &= \int L(S, \theta_S|\mathbf{D}) P(\theta_S|S) d\theta_S \end{aligned}$$

  Hence it is called the **marginal likelihood** of S and is denoted as $L(S|\mathbf{D})$. $logL(S|\mathbf{D})$ is denoted as $l(S|\mathbf{D})$.

- Notes:
  1. $P(\theta_S|S)$ is the parameter prior.
  2. The marginal loglikelihood $l(S|\mathbf{D})$ is NOT the same as the maximized loglikelihood $l^*(S|\mathbf{D})$.

Marginal Likelihood

Marginal likelihood has closed-form under the following assumptions:

1. Data **D** are random i.i.d samples from some (unknown) BN.
2. All cases in **D** are complete.
3. For each structure S, the parameter prior $p(\theta_S|S)$
   1. Satisfies the parameter (global and local) independence assumption.
   2. Is the product Dirichlet distribution:

$$p(\theta_S|S) \propto \prod_{i,k} \prod_{j} \theta_{ijk}^{\alpha_{ijk}-1}$$

We call these assumptions **Cooper and Herskovits (CH)** assumptions.

Marginal Likelihood

Theorem (8.1)

(Cooper and Herskovits (1992)) *Under the CH assumptions,*

$$logP(\mathbf{D}|S) = \sum_{i,k}[log\frac{\Gamma(\alpha_{i*k})}{\Gamma(\alpha_{i*k}+m_{i*k})} + \sum_{j} log\frac{\Gamma(\alpha_{ijk}+m_{ijk})}{\Gamma(\alpha_{ijk})}]$$

where

- m_{ijk}: *number of data cases where* X_i=j *and* $pa_S(X_i) = k$.
- $m_{i*k} = \sum_j m_{ijk}$: *number of data cases where* $pa_S(X_i) = k$.
- $\alpha_{i*k} = \sum_j \alpha_{ijk}$.

This is sometimes called the **Cooper-Herskovits (CH) scoring function**, or the **Bayesian Dirichlet equivalence (BDe) score**.

CH Scoring Function

- How to choose the α_{ijk}?
- Not an easy task since we need to do this for all structures. There are lots of them!
- One solution:
  - Equivalent sample size: α
  - A BN $\mathcal{N}_0$ that represent prior joint probability $P_0(X_1, X_2, \ldots, X_n)$.
  - Set $\alpha_{ijk} = \alpha * P_0(X_i{=}j|pa_S(X_i){=}k)$.
  - $P_0(X_i{=}j|pa_S(X_i){=}k)$ can be computed via standard BN inference (Clique tree propagation.)
- Note: Sometimes, it is natural for different models to have different equivalent sample sizes (Kayaalp an Cooper, UAI02).

Choice of structure prior

Choice of structure prior $P(S)$

- Can just be uniform for convenience.
- Exclude impossible structures (based on judgment of causal relationships) and impose a uniform prior on the set of remain structures.
  - Note that this could compromise the optimality of search. It might happen that the only way to the optimal model is through some impossible models.
- Or impose an order on the variables (structures are then limited) and then use uniform prior.
- ...

Introduction

- We next derive asymptotic (large sample) approximation of the marginal likelihood
  - Bayesian score is asymptotically the same as the marginal likelihood provide parameter prior is positive everywhere.
- Why interesting?
  - Leading to model selection criteria that can be used even when the CH assumptions are not true.
  - Allowing us to study the asymptotic properties of the marginal likelihood.

Two Assumptions

- Simplifying notation: change θ_S to θ. View it as a column vector.
- Let θ^* be the ML estimate of θ:

$$\theta^*_{ijk} = \frac{m_{ijk}}{m_{i*k}}$$

- **Assumption 1**:$P(\mathbf{D}|S,\theta)$ has a unique maximum point θ^*. In other words, for any $\theta \neq \theta^*$,

$$P(\mathbf{D}|S,\theta) < P(\mathbf{D}|S,\theta^*)$$

- **Assumption 2**: The ML estimation θ^* is an interior point in the parameter space. In other words, $\theta^*_{ijk} > 0$ for all i, j, and k.
- In additional, assume complete data (although result is used also in the case of incomplete data).

A Property of Loglikelihood Function

- Consider the loglikelihood function

$$\begin{aligned} l(S,\theta|\mathbf{D}) &= \sum_{i,k}\sum_{j} m_{ijk} log\theta_{ijk} \\ &= \sum_{i,k} m_{i*k} \sum_{j} \theta^*_{ijk} log\theta_{ijk} \\ &= \sum_{i,k} m_{i*k} [\sum_{j} \theta^*_{ijk} log\frac{\theta_{ijk}}{\theta^*_{ijk}} + \sum_{j} \theta^*_{ijk} log\theta^*_{ijk}] \end{aligned}$$

- Hence

$$P(\mathbf{D}|S,\theta) = exp\{l(S,\theta|\mathbf{D})\} = \prod_{i,k} (exp\{\sum_{j} \theta^*_{ijk} log\frac{\theta_{ijk}}{\theta^*_{ijk}} + \sum_{j} \theta^*_{ijk} log\theta^*_{ijk}\})^{m_{i*k}}$$

A Property of Loglikelihood Function

$$P(\mathbf{D}|S,\theta) = \prod_{i,k}(exp\{\sum_j \theta^*_{ijk} log\frac{\theta_{ijk}}{\theta^*_{ijk}} + \sum_j \theta^*_{ijk} log\theta^*_{ijk}\})^{m_{i*k}}$$

- As a function of θ, $P(\mathbf{D}|S,\theta)$ reaches the maximum at θ^*.
- When the sample size is large, m_{i*k} is also large.
- Hence, as θ moves away from θ^*, $P(\mathbf{D}|S,\theta)$ decreases quickly.
- Now consider, $P(\mathbf{D}|S) = \int P(\mathbf{D}|S,\theta)P(\theta|S)d\theta$.
- It can approximated by performing the integration in a small neighborhood of θ^*.
- This leads to the Laplace approximation.

Deriving the Laplace Approximation

- For simplicity, denote $l(S, \theta|\mathbf{D})$ as $l(\theta)$.
- Since θ^* maximizes $l(\theta)$,

$$l'(\theta^*) = \mathbf{0}$$

- Use Taylor serious expansion of $l(\theta)$ around θ^*, we get that, in a small neighborhood of θ^*,

$$l(\theta) \approx l(\theta^*) + \frac{1}{2}(\theta - \theta^*)^T l''(\theta^*)(\theta - \theta^*)$$

where $l''(\theta^*)$ is the Hessian matrix of l evaluated at θ^*:

$$l''(\theta^*) = [\frac{\partial^2 l(\theta)}{\partial\theta_{ijk}\partial\theta_{abc}}]_{\theta=\theta^*}$$

Deriving the Laplace Approximation

- Let $A = -l''(\theta^*)$. In a small neighborhood of θ^*,

$$l(\theta) \approx l(\theta^*) - \frac{1}{2}(\theta - \theta^*)^T A`(\theta - \theta^*)$$

-

- $P(\mathbf{D}|S, \theta) = exp\{l(\theta)\}$:
  - $\approx exp\{l(\theta^*) - \frac{1}{2}(\theta - \theta^*)^T A(\theta - \theta^*)\}$ in a small neighborhood around θ^*.
  - ≈ 0 outside the neighborhood.

Deriving the Laplace Approximation

- Since $l(\theta^*) > l(\theta)$ for any $\theta \neq \theta^*$, $A = -l''(\theta^*)$ is positive-definite.
- Let d be the number of free parameters in S.
- It is known that

$$|A| = O(d log N)$$

- Hence
  - $exp\{l(\theta^*) - \frac{1}{2}(\theta - \theta^*)^T A(\theta - \theta^*)\}$ is close to 0 except in a small neighborhood of θ^*
- Therefore, $P(\mathbf{D}|S, \theta) \approx exp\{l(\theta^*) - \frac{1}{2}(\theta - \theta^*)^T A(\theta - \theta^*)\}$ everywhere.

Deriving the Laplace Approximation

- Now consider the marginal likelihood:

$$\begin{aligned} P(\mathbf{D}|S) &= \int P(\mathbf{D}|S,\theta)P(\theta|S)d\theta \\ &\approx \int \exp\{l(\theta^*) - \frac{1}{2}(\theta-\theta^*)^T A(\theta-\theta^*)\}P(\theta|S)d\theta \\ &= \exp\{l(\theta^*)\}\int \exp\{-\frac{1}{2}(\theta-\theta^*)^T A(\theta-\theta^*)\}P(\theta|S)d\theta \\ &\approx P(\mathbf{D}|S,\theta^*)P(\theta^*|S)\int \exp\{-\frac{1}{2}(\theta-\theta^*)^T A(\theta-\theta^*)\}d\theta \end{aligned}$$

The last step is due to the fact that the integrand is small except in a small neighborhood of θ^*.

Deriving the Laplace Approximation

- Note that $\frac{1}{\sqrt{(2\pi)^d|A|^{-1}}}\exp\{\frac{1}{2}(\theta-\theta^*)^T A(\theta-\theta^*)\}$ is the Gaussian distribution with covariance matrix A.
- Hence

$$P(\mathbf{D}|S) \approx P(\mathbf{D}|S,\theta^*)P(\theta^*|S)\sqrt{(2\pi)^d|A|^{-1}}$$

  where d is the number of free parameters S or in the vector θ.
- The log marginal likelihood:

$$logP(\mathbf{D}|S) \approx logP(\mathbf{D}|S,\theta^*) + logP(\theta^*|S) + \frac{d}{2}log(2\pi) - \frac{1}{2}log|A|$$

  Note that the first term is the maximized loglikelihood.
- This is known as the **Laplace approximation**.

Laplace Approximation

- Kass et al (1988) showed that, under certain conditions (two of which given as assumptions above),

$$\frac{P(\mathbf{D}|S) - P(\mathbf{D}|S)_{\text{Laplace}}}{P(\mathbf{D}|S)} = O(1/N)$$

  where probability 1. Hence it is extremely accurate.

The BIC Score

- In the Laplace approximation, the two terms in the middle do not increase with N.
- If we ignore those two terms and approximate $log|A|$ using $dlogN$, we get the **Bayesian information criterion (BIC)**:

$$logP(\mathbf{D}|S) \approx logP(\mathbf{D}|S,\theta^*) - \frac{d}{2}logN$$

- Quality of approximation: $O(1)$ (Schwarz 1978, Haughton 1988, Kass and Wasserman 1995, Raftery 1995).

The BIC Score

$$logP(\mathbf{D}|S) \approx logP(\mathbf{D}|S,\theta^*) - \frac{d}{2}logN$$

- The first term of the BIC score is the maximized loglikelihood. It measures model fit.
- The second term penalizes model complexity.
- **This avoids overfitting**.
  - The Bayesian score does not lead to overfitting.
- BIC is one example of **penalized likelihood** (Lanternman 2001).
- Maximized loglikelihood increases linearly with sample size, while the penalty term increase logarithmically.
  - More and more emphasis is placed on model fit as sample size increases.

MDL

The **minimum description length (MDL)** score (Rissanen 1987):

- Machine learning is about finding regularities in data.
- Regularities should allow us to describe the data concisely.
- Find model to minimize

$$\text{Description length of model} + \text{Description length of data}$$

- It turns out to be the negation of the BIC score.
- Description length of data is related to likelihood as illustrated in Huffman's coding.

AIC

- **Akaike information criterion**:
- Idea:
  - **D** sampled from $P(\mathbf{X})$.
  - Based on **D**, find $\mathcal{N}^* = (S^*, \theta^*)$ such that

$$KL(P, P_{\mathcal{N}^*}) \leq KL(P, P_{\mathcal{N}}), \forall \mathcal{N}$$

    (Note: the complete model does not necessarily minimize the KL due to overfitting.)
  - Under certain conditions, S^* should maximize the AIC score:

$$AIC(S|\mathbf{D}) = \log P(\mathbf{D}|S, \theta^*) - d$$

- Models obtained using AIC typically are more complex than those obtained using BIC.

Holdout validation and cross validation

- **Holdout validation**:
  - Split data into **training set** and **validation set**.
  - Parameter estimation based on training set.
  - Model score: likelihood based on validation set.
- **Cross validation**:
  - Split data into k subsets
  - Use each subset as validation set and the rest as training set, and obtains a score.
  - Total model score: average of the scores for all the cases.
- Both are equivalent to AIC asymptotically.

Model Inclusion and Equivalence

- A model S **includes** a joint distribution $P(\mathbf{X})$
  - if there is a parameter vector θ such that (S, θ) represents $P(\mathbf{X})$.
- If S includes P and all other models that include $P(\mathbf{X})$ have the same number or more parameters than S, then S is said to be a **parsimonious** model (wrt P).
- One model S **includes** another model S', if it includes all the joint distributions that S' can represent.
- If S includes S', and vice versa, then S and S' are said to be **distributionally equivalent**.
- If two distributionally equivalent models have the same number of parameters, then they are **equivalent**.

Consistency

- **D** sample from $P(\mathbf{X})$.
- A scoring function f is **consistent** if, when the sample size goes to infinite, the following two conditions are satisfied:

1. If S includes P and S' does not include P, then,

$$f(S|\mathbf{D}) > f(S'|\mathbf{D})$$

2. If both S and S' includes P, and S has fewer parameters than S' then,

$$f(S|\mathbf{D}) > f(S'|\mathbf{D})$$

Consistency

- Suppose $\mathbf{D}$ is sampled from $P(\mathbf{X})$ represented by (S, θ) and S is parsimonious.
- Further suppose that there does not exist S' that includes P, has the same number of parameters as S, but is not equivalent to S.
- If f is consistent, then, when the sample size goes to infinite,

$$f(S|\mathbf{D}) > f(S'|\mathbf{D})$$

  for all other model S' that is not equivalent to S.
- Hence, we can in principle recover the generative model (S, θ) from data.

Consistency

- Consistent scoring functions:
  - Bayesian score, marginal likelihood, BDE, BIC, MDL
- Inconsistent scoring functions:
  - AIC, holdout validation, cross validation.

Outline

The Problem

- Model optimization:How to find the structure that maximizes a scoring function?
- A naive method: Exhaustive search
  - Compute the score of every structure
  - Pick the one with the highest score.

Number of Possible Structures

- $f(n)$: the number of unlabeled DAGs on n nodes.
- Robinson (1977) [1] showed that

$$\begin{aligned} f(1) &= 1 \\ f(n) &= \sum_{i=1}^{n} (-1)^{i+1} \frac{n!}{(n-i)!\,i!} 2^{i(n-i)} f(n-i) \end{aligned}$$

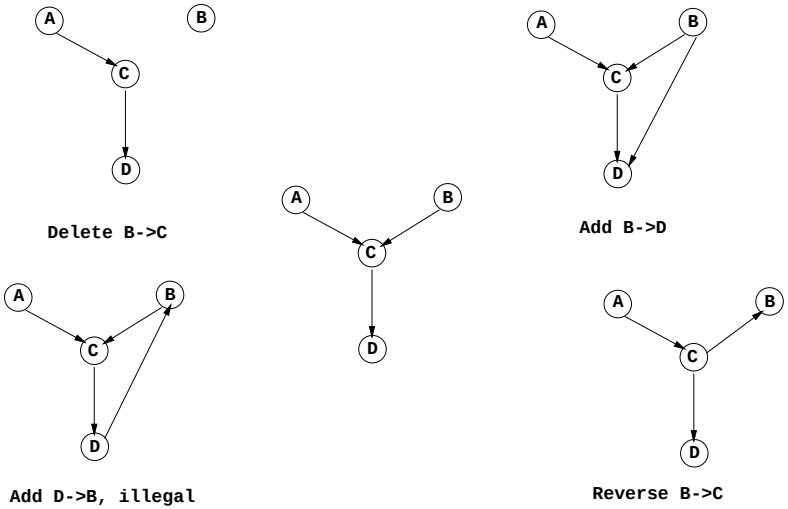
  - No closed form is known.
  - $f(10) \approx 4.2 \times 10^{18}$
- BNs are labeled DAGs.
  The number of BN structures for n variables is larger than $f(n)$.
- Exhaustive search is infeasible.

[1] Robinson, R. W. (1977). Counting unlabelled acyclic digraphs. In *Lecture Notes in Mathematics: Combinatorial Mathematics V*, (ed. C. H. C. Little). Springer-Verlag, New York.

Hill Climbing

- Start with an initial structure.
- Repeat until termination:
  - Generate a set of structures by modifying the current structure.
  - Compute their scores.
  - Pick the one with the highest score and use it as the current model in the next step.
  - Terminate when model score cannot is not improved.
- Return the best network.

Search Operators



Search operators for modifying a structure:

- Add an arc.
- Delete an arc.
- Reverse an arc.

Note:

- The add-arc and reverse-arc not permitted if results in directed cycles.

Evaluating Candidate Models

- Suppose there are n variables.
- The number of candidate models at each iteration: $O(n^2)$.
- We need to compute the score of each of the candidate models.
- This is the most time-consuming step.
- Structures of scoring functions can be exploited to simplify the computation.

Decomposition of Scoring Functions

- Some coring functions **decompose** according to the variables and each component depends only on a variable and its parents.
- The CH score:

$$\begin{aligned} CH(S|\mathbf{D}) &= \sum_{\mathbf{i}} \sum_{k} [log \frac{\Gamma(\alpha_{i*k})}{\Gamma(\alpha_{i*k} + m_{i*k})} + \sum_{j} log \frac{\Gamma(\alpha_{ijk} + m_{ijk})}{\Gamma(\alpha_{ijk})}] \\ &= \sum_{i} CH(X_i, pa_S(X_i)|\mathbf{D}) \end{aligned}$$

where the **family score**

$$CH(X_i, pa_S(X_i)|\mathbf{D}) = \sum_{k} [log \frac{\Gamma(\alpha_{i*k})}{\Gamma(\alpha_{i*k} + m_{i*k})} + \sum_{j} log \frac{\Gamma(\alpha_{ijk} + m_{ijk})}{\Gamma(\alpha_{ijk})}]$$

depends only on X_i and its parents in S.

Decomposition of Scoring Functions

- The BIC score:

$$\begin{aligned} BIC(S|\mathbf{D}) &= logP(\mathbf{D}|S,\theta^*) - \frac{d}{2}logN \\ &= \sum_i \sum_k \sum_j m_{ijk} log \frac{m_{ijk}}{\sum_j m_{ijk}} - \sum_i \frac{q_i(r_i-1)}{2} logN \\ &= \sum_i BIC(X_i, pa_S(X_i)|\mathbf{D}) \end{aligned}$$

where the **family score**

$$BIC(X_i, pa_S(X_i)|\mathbf{D}) = \sum_k \sum_j m_{ijk} log \frac{m_{ijk}}{\sum_j m_{ijk}} - \frac{q_i(r_i-1)}{2} logN$$

where q_i is the number of states of the parents of X_i and r_i is the number of states of X_i.

Evaluating Candidate Model

- Suppose S is the current model and we have computed $BIC(S|\mathbf{D})$.
- Suppose candidate model S' is obtained from S by adding an arc from some node to X_i.
- Then

$$BIC(S'|\mathbf{D}) - BIC(S|\mathbf{D}) = BIC(X_i, \pi'(X_i)|\mathbf{D}) - BIC(X_i, \pi(X_i)|\mathbf{D})$$

- Hence we can compute $BIC(S'|\mathbf{D})$ efficiently using:

$$BIC(S'|\mathbf{D}) = BIC(S|\mathbf{D}) + BIC(X_i, \pi'(X_i)|\mathbf{D}) - BIC(X_i, \pi(X_i)|bfD)$$

Only local counting are involved.

Initial Structure

- Empty network: Network with no arcs.
- Network built using heuristics.
- Random network.

Problems with Hill Climbing

- Local maxima:
  All one-edge changes reduced the score, but not optimal yet.
- Plateaus:
  Neighbors have the same score.

Solutions:

- Random restart.
- TABU-search:
  - Keep a list of K most recently visited structures and avoid them.
  - Avoid plateau.
- Simulated annealing.

Outline

Model Selection with Incomplete Data

- The CH score is not applicable in the case of incomplete data.
- We will use the BIC score:

$$BIC(S|\mathbf{D}) \quad = \quad logP(\mathbf{D}|S,\theta^*) - \frac{d}{2}logN$$

- No longer have:

$$logP(\mathbf{D}|S,\theta^*) - \frac{d}{2}logN = \sum_i \sum_k \sum_j m_{ijk} log \frac{m_{ijk}}{\sum_j m_{ijk}}$$

- But we can compute $logP(\mathbf{D}|S,\theta^*)$ using EM.

Straightforward Model Evaluation

- At each step, we need to evaluate $O(n^2)$ candidate models.
- The BIC scores cannot be computed by simple counting.
- Iterative algorithms, mostly EM, are used.
- To compute the BIC score of EACH candidate model, we need to run EM to estimate θ^*.
- EM requires BN inference, once for EACH iteration.
- EM takes a large (hundreds) number of iterations to converge.
- Computationally prohibitive.

The Idea of Structural EM

The Idea:

- Find ML estimate θ^* of the parameters for the current model S.
- Complete data using the current model (S, θ^*).
- Evaluate candidate models using the completed data.

Advantage:

- Instead of running EM on all candidate structures, we run EM only on ONE structure, namely the current structure.

BIC Score of Structure and Parameters

- The BIC score $BIC(S|\mathbf{D})$ is a function of structures.
- For convenience, define

$$BIC(S, \theta|\mathbf{D}) = logP(\mathbf{D}|S, \theta) - \frac{d(S)}{2} logN$$

  Measures how good the model S is when its parameters are θ.
- Then

$$BIC(S|\mathbf{D}) = BIC(S, \theta^*|\mathbf{D})$$

Best Structure and Parameters

- Want S^* and θ^* such that

$$BIC(S^*, \theta^* | \mathbf{D}) \geq BIC(S, \theta | \mathbf{D})$$

  for any BN (S, θ).

- Question:
  - Suppose $(\bar{S}, \bar{\theta})$ be the current BN.
  - How to find another BN (S, θ) that increases the BIC score?

$$BIC(S, \theta | \mathbf{D}) > BIC(\bar{S}, \bar{\theta} | \mathbf{D})$$

- Problem solved if we know how to do this cheaply.
- This is what we want.

Expected BIC Score

- This is what we do:
- Let $\bar{\mathbf{D}}$ be the completion of $\mathbf{D}$ by $(\bar{S}, \bar{\theta})$.

$$BIC(S, \theta|\bar{\mathbf{D}}) = \sum_I \sum_{\mathbf{X}_I} P(\mathbf{X}_I|D_I, \bar{S}, \bar{\theta}) logP(D_I, \mathbf{X}_I|S, \theta) - \frac{d(S)}{2} logN,$$

- It is the expected value of $BIC(S, \theta|\mathbf{D})$, where the expectation is takes w.r.t $(\bar{S}, \bar{\theta})$.

BIC and Expected BIC

- We will denote the expected BIC score $BIC(S, \theta|\bar{\mathbf{D}})$ by $Q(S, \theta|\bar{S}, \bar{\theta})$:

$$Q(S, \theta|\bar{S}, \bar{\theta}) = \sum_I \sum_{\mathbf{X}_I} P(\mathbf{X}_I|D_I, \bar{S}, \bar{\theta}) logP(D_I, \mathbf{X}_I|S, \theta) - \frac{d(S)}{2} logN,$$

- Similar to Lecture 7, we have

$$\begin{aligned} BIC(S, \theta|\mathbf{D}) &= logP(\mathbf{D}|S, \theta) - \frac{d(S)}{2} logN \\ &= \sum_I \sum_{\mathbf{X}_I} P(\mathbf{X}_I|D_I, \bar{S}, \bar{\theta}) logP(D_I, \mathbf{X}_I|S, \theta) \\ &\quad - \sum_I \sum_{\mathbf{X}_I} P(\mathbf{X}_I|D_I, \bar{S}, \bar{\theta}) logP(\mathbf{X}_I|D_I, S, \theta) - \frac{d(S)}{2} logN \\ &= Q(S, \theta|\bar{S}, \bar{\theta}) - \sum_I \sum_{\mathbf{X}_I} P(\mathbf{X}_I|D_I, \bar{S}, \bar{\theta}) logP(\mathbf{X}_I|D_I, S, \theta). \end{aligned}$$

BIC Score and Expected BIC Score

$$BIC(S,\theta|\mathbf{D}) = Q(S,\theta|\bar{S},\bar{\theta}) - \sum_{I}\sum_{\mathbf{X}_I} P(\mathbf{X}_I|D_I,\bar{S},\bar{\theta}) log P(\mathbf{X}_I|D_I,S,\theta)$$

$$BIC(\bar{S},\bar{\theta}|\mathbf{D}) = Q(\bar{S},\bar{\theta}|\bar{S},\bar{\theta}) - \sum_{I}\sum_{\mathbf{X}_I} P(\mathbf{X}_I|D_I,\bar{S},\bar{\theta}) log P(\mathbf{X}_I|D_I,\bar{S},\bar{\theta})$$

Theorem (8.2)

- *If*

$$Q(S,\theta|\bar{S},\bar{\theta}) > Q(\bar{S},\bar{\theta}|\bar{S},\bar{\theta})$$

- *then*

$$BIC(S,\theta|\mathbf{D}) > BIC(\bar{S},\bar{\theta}|\mathbf{D})$$

Improving Expected BIC Score

Problem becomes:

- How to find (S, θ) such that $Q(S, \theta|\bar{S}, \bar{\theta}) > Q(\bar{S}, \bar{\theta}|\bar{S}, \bar{\theta})$?
- Method 1:Improve the parameters

$$Q(\bar{S}, \theta|\bar{S}, \bar{\theta}) > Q(\bar{S}, \bar{\theta}|\bar{S}, \bar{\theta})$$

- Parameter estimation in the case of complete data.Computationally cheap.

Improving Expected BIC Score

- Method 2: Improve the model structure (together with parameters):

$$Q(S, \theta | \bar{S}, \bar{\theta}) > Q(\bar{S}, \bar{\theta} | \bar{S}, \bar{\theta})$$

- Structure learning in the case of complete data. Computationally cheap.

The Structural EM Algorithm

- Pick initial structure S^0 and initial parameters $\theta^{0,0}$.
- For $t = 0$ to ∞
  1. **Improve parameters**: For $r = 0, 1, 2, \ldots$ until convergence or some $r$max
     - **Standard parametric EM step**:

$$\theta^{t,r+1} = \arg\max_{\theta} Q(S^t, \theta | S^t, \theta^{t,r})$$

  2. **Improve Structure**:
     - Generate candidate structures by modifying S_t using the search operators.
     - Let S^{t+1} be the candidate structure that maximizes

$$\max_{\theta} Q(S^{t+1}, \theta | S^t, \theta^{t,r}), \qquad \text{and}$$

$$\theta^{t+1,0} = \arg\max_{\theta} Q(S^{t+1}, \theta | S^t, \theta^{t,r})$$

     - If $BIC(S^{t+1}, \theta^{t+1,0}|\mathbf{D}) \leq BIC(S^t, \theta^{t,r}|\mathbf{D}) + \epsilon$, return BN $(S^t, \theta^{t,r})$.

Convergence

- When improving parameters, we have

$$Q(S^t, \theta^{t,r+1}|S^t, \theta^{t,r}) > Q(S^t, \theta^{t,r}|S^t, \theta^{t,r})$$

.

By Theorem 8.2, this implies

$$BIC(S^t, \theta^{t,r+1}|\mathbf{D}) > Q(S^t, \theta^{t,r}|\mathbf{D})$$

- When improving structure, we have

$$Q(S^{t+1}, \theta^{t+1,0}|S^t, \theta^{t,r}) > Q(S^t, \theta^{t,r}|S^t, \theta^{t,r})$$

By Theorem 8.2, this implies

$$BIC(S^{t+1}, \theta^{t+1,0}|\mathbf{D}) > Q(S^t, \theta^{t,r}|\mathbf{D})$$

- So $BIC(S^t, \theta^{t,r}|\mathbf{D})$ increases monotonically with t and r.
- Hence structural EM converges.

Converges to What?

- The parametric part converges to global or local parametric maxima or saddle points in the parameter space.
- Structure? No much to say.Converges to what?
- Empirical results indicates that structural EM finds good structures.