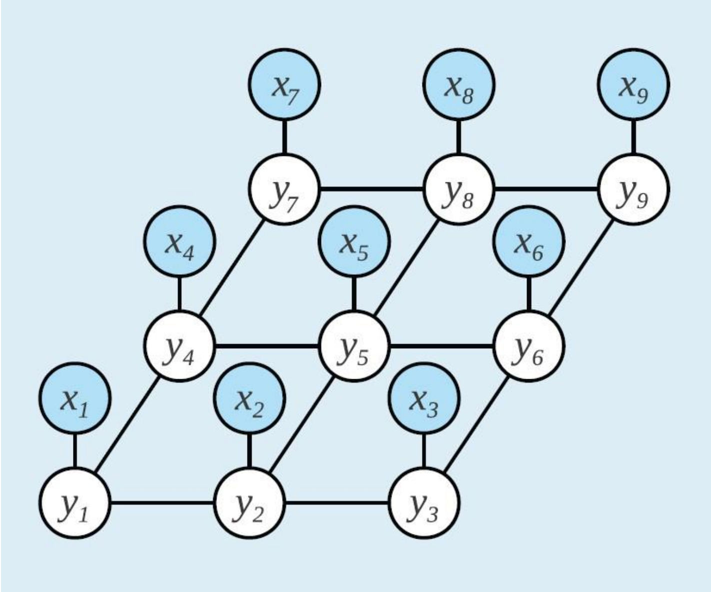
# Probabilistic Graphical Models in Bioinformatics

Lecture 8: score-based structure learning



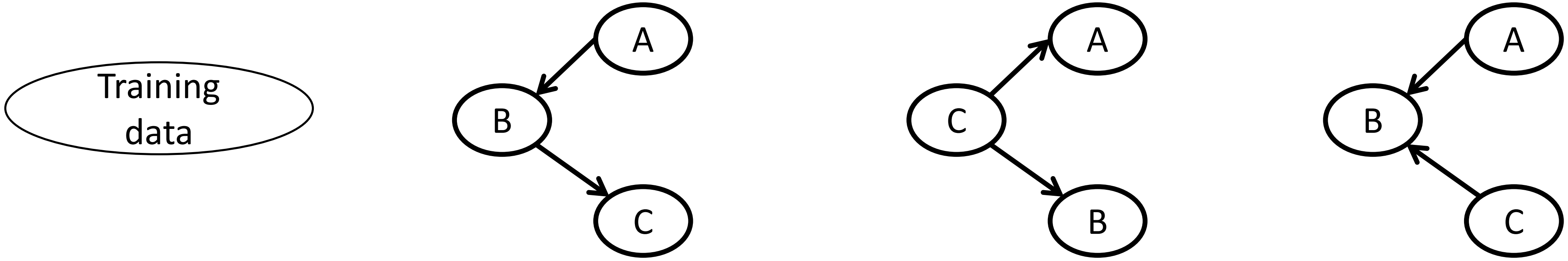


Hesam Montazeri
Department of Bioinformatics, IBB, University of Tehran


Ordibehesht 4, 1398
(April 24, 2019)

# Score-based structure learning

- Define a **scoring function** that can score each candidate structure with respect to the training data.
- Search for a high-scoring structure




- Outline
  - Structure scores
    - Likelihood score
    - Bayesian score
  - Structure search

# Likelihood score

- Find a graph $g$ and parameters $\theta_g$ that maximizes the likelihood. Hence, we have

$$\begin{aligned}\max_{\mathcal{G},\boldsymbol{\theta}_{\mathcal{G}}} L(\langle \mathcal{G}, \boldsymbol{\theta}_{\mathcal{G}} \rangle : \mathcal{D}) &= \max_{\mathcal{G}}[\max_{\boldsymbol{\theta}_{\mathcal{G}}} L(\langle \mathcal{G}, \boldsymbol{\theta}_{\mathcal{G}} \rangle : \mathcal{D})] \\ &= \max_{\mathcal{G}}[L(\langle \mathcal{G}, \hat{\boldsymbol{\theta}}_{\mathcal{G}} \rangle : \mathcal{D})].\end{aligned}$$

- We should find the graph structure $g$ that achieves the highest likelihood when we use the MLE for parameters of $g$.

$$\mathrm{score}_L(\mathcal{G} : \mathcal{D}) = \ell(\hat{\boldsymbol{\theta}}_{\mathcal{G}} : \mathcal{D}),$$

# Simple example

- Consider the model $g_0$ where $X$ and Y are independent. **Question:** what is the likelihood score for $g_0$?

$$\text{score}_L(\mathcal{G}_0 \ : \ \mathcal{D}) = \sum_m \log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]}.$$

- Consider the model $g_1 : X \to Y$. **Question:** what is the likelihood score for $g_1$?

$$\text{score}_L(\mathcal{G}_1 \ : \ \mathcal{D}) = \sum_m \log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]|x[m]},$$

- Compute the difference of two models:

$$\text{score}_L(\mathcal{G}_1 \ : \ \mathcal{D}) - \text{score}_L(\mathcal{G}_0 \ : \ \mathcal{D}) = \sum_m \log \hat{\theta}_{y[m]|x[m]} - \log \hat{\theta}_{y[m]}.$$

- Some simple algebra

$$\text{score}_L(\mathcal{G}_1 \ : \ \mathcal{D}) - \text{score}_L(\mathcal{G}_0 \ : \ \mathcal{D}) = \sum_{x,y} M[x,y] \log \hat{\theta}_{y|x} - \sum_y M[y] \log \hat{\theta}_y.$$

$$\text{score}_L(\mathcal{G}_1 \ : \ \mathcal{D}) - \text{score}_L(\mathcal{G}_0 \ : \ \mathcal{D}) = M \sum_{x,y} \hat{P}(x,y) \log \frac{\hat{P}(y \mid x)}{\hat{P}(y)} = M \cdot \boldsymbol{I}_{\hat{P}}(X;Y),$$

# General decomposition

- **Proposition:** the *likelihood score* decomposes as follows

$$\mathrm{score}_L(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^{n} \boldsymbol{I}_{\hat{P}}(X_i; \mathrm{Pa}_{X_i}^{\mathcal{G}}) - M \sum_{i=1}^{n} \boldsymbol{H}_{\hat{P}}(X_i).$$

where

$$\mathbf{I}_P(\boldsymbol{X};\boldsymbol{Y}) = \sum_{\boldsymbol{x},\boldsymbol{y}} P(\boldsymbol{x},\boldsymbol{y}) \log \frac{P(\boldsymbol{x},\boldsymbol{y})}{P(\boldsymbol{x})P(\boldsymbol{y})}$$

$$\boldsymbol{H}_P(\boldsymbol{X}) = -\sum_{\boldsymbol{x}} P(\boldsymbol{x}) \log P(\boldsymbol{x})$$

- **Decomposable score:** a structure score is decomposable if the score can be written as

$$\mathrm{score}(\mathcal{G} : \mathcal{D}) = \sum_i \mathrm{FamScore}(X_i \mid \mathrm{Pa}_{X_i}^{\mathcal{G}} : \mathcal{D}),$$

- Example:

$$\mathrm{FamScore}_L(X \mid \boldsymbol{U} : \mathcal{D}) = M \cdot \left[ \boldsymbol{I}_{\hat{P}}(X;\boldsymbol{U}) - \boldsymbol{H}_{\hat{P}}(X) \right].$$

# General decomposition-proof

$$\text{score}_L(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^{n} \boldsymbol{I}_{\hat{P}}(X_i; \text{Pa}_{X_i}^{\mathcal{G}}) - M \sum_{i=1}^{n} \boldsymbol{H}_{\hat{P}}(X_i).$$

- We write likelihood as follows

$$\ell(\hat{\boldsymbol{\theta}}_{\mathcal{G}} : \mathcal{D}) = \sum_{i=1}^{n} \left[ \sum_{\boldsymbol{u}_i \in Val(\text{Pa}_{X_i}^{\mathcal{G}})} \sum_{x_i} M[x_i, \boldsymbol{u}_i] \log \hat{\theta}_{x_i | \boldsymbol{u}_i} \right].$$

- For each term in the square bracket

$$\begin{aligned}
&\frac{1}{M} \sum_{\boldsymbol{u}_i} \sum_{x_i} M[x_i, \boldsymbol{u}_i] \log \hat{\theta}_{x_i | \boldsymbol{u}_i} \\
&= \sum_{\boldsymbol{u}_i} \sum_{x_i} \hat{P}(x_i, \boldsymbol{u}_i) \log \hat{P}(x_i \mid \boldsymbol{u}_i) \\
&= \sum_{\boldsymbol{u}_i} \sum_{x_i} \hat{P}(x_i, \boldsymbol{u}_i) \log \left( \frac{\hat{P}(x_i, \boldsymbol{u}_i)}{\hat{P}(\boldsymbol{u}_i)} \frac{\hat{P}(x_i)}{\hat{P}(x_i)} \right) \\
&= \sum_{\boldsymbol{u}_i} \sum_{x_i} \hat{P}(x_i, \boldsymbol{u}_i) \log \frac{\hat{P}(x_i, \boldsymbol{u}_i)}{\hat{P}(\boldsymbol{u}_i)\hat{P}(x_i)} + \sum_{x_i} \left( \sum_{\boldsymbol{u}_i} \hat{P}(x_i, \boldsymbol{u}_i) \right) \log \hat{P}(x_i) \\
&= \boldsymbol{I}_{\hat{P}}(X_i; \boldsymbol{U}_i) - \sum_{x_i} \hat{P}(x_i) \log \frac{1}{\hat{P}(x_i)} \\
&= \boldsymbol{I}_{\hat{P}}(X_i; \boldsymbol{U}_i) - \boldsymbol{H}_{\hat{P}}(X_i),
\end{aligned}$$

# Limitations of the maximum likelihood score

- Score maximizes for the fully connected graph

$$\text{score}_L(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^{n} \boldsymbol{I}_{\hat{P}}(X_i; \text{Pa}^{\mathcal{G}}_{X_i}) - M \sum_{i=1}^{n} \boldsymbol{H}_{\hat{P}}(X_i).$$

- Mutual information is equal 0 if and only if X and Y are independent
    - Almost never happen in empirical distribution $\hat{P}$.

- In other words, likelihood score overfits the data.

- How to avoid overfitting:
    - Disallow complex models: restrict #parents or #parameters
    - Or using scores that penalize complexity

# Bayesian score

# Bayesian score

University of TEHRAN

- An alternative scoring function based on a Bayesian perspective
- Main principle of the Bayesian approach:
  - Since we have uncertainty over both structure and parameters, we define a structure prior $P(g)$ and a parameter prior $P(\theta_g \mid g)$

$$P(\mathcal{G} \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \mathcal{G})P(\mathcal{G})}{P(\mathcal{D})},$$

- We define the *Bayesian score* as:

$$\text{score}_B(\mathcal{G} \ : \ \mathcal{D}) = \log P(\mathcal{D} \mid \mathcal{G}) + \log P(\mathcal{G}).$$

Almost irrelevant compared to the first term

- $P(D \mid g)$ is called the marginal likelihood of the data given structure

$$P(\mathcal{D} \mid \mathcal{G}) = \int_{\Theta_{\mathcal{G}}} P(\mathcal{D} \mid \boldsymbol{\theta}_{\mathcal{G}}, \mathcal{G}) P(\boldsymbol{\theta}_{\mathcal{G}} \mid \mathcal{G}) d\boldsymbol{\theta}_{\mathcal{G}},$$

# Marginal likelihood

- The marginal likelihood is quite different from the maximum likelihood score.
- Both examine the likelihood of the data given the structure
- **Maximum likelihood score:** returns maximum of this function
- **Marginal likelihood:** average value of this function based on parameter prior $P(\theta_g \mid g)$
- Marginal likelihood avoids overfitting because it is not sensitive to a particular choice of parameters.
- **Another motivation:** marginal likelihood can be viewed as a score evaluates the ability of the model to predict a new data instances

$$P(\mathcal{D} \mid \mathcal{G}) = \prod_{m=1}^{M} P(\xi[m] \mid \xi[1], \ldots, \xi[m-1], \mathcal{G}).$$

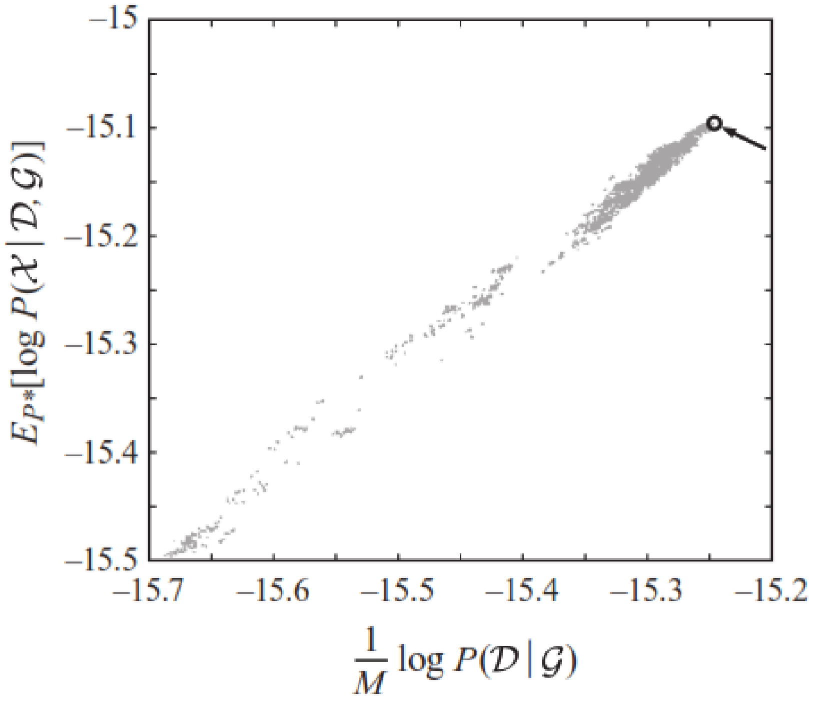
Probability of the *m*'th instance given the first *m-1* instances (Bayesian prediction)

# Marginal likelihood-2

- This intuition suggests that $P(D \mid g)$ is an estimator for the average log-likelihood of a new sample from the distribution $P^*$

$$\frac{1}{M} \log P(\mathcal{D} \mid \mathcal{G}) \approx \boldsymbol{E}_{P^*}[\log P(\mathcal{X} \mid \mathcal{G}, \mathcal{D})]$$

Expected likelihood of new samples




Marginal likelihood in training data

# Marginal likelihood for a single variable

- Consider a single binary random variable X, a dataset with M[1] heads and M[0] tails, and a prior distribution $Dirichlet(\alpha_1, \alpha_0)$
- ML score:

$$P(\mathcal{D} \mid \hat{\theta}) = \left(\frac{M[1]}{M}\right)^{M[1]} \cdot \left(\frac{M[0]}{M}\right)^{M[0]}$$

- Marginal likelihood:

$$P(x[1], \ldots, x[M]) = P(x[1]) \cdot P(x[2] \mid x[1]) \cdot \ldots \cdot P(x[M] \mid x[1], \ldots, x[M-1]).$$

Recall: $$P(x[m+1] = H \mid x[1], \ldots, x[m]) = \frac{M^m[1] + \alpha_1}{m + \alpha},$$

so
$$\begin{aligned} P(x[1], \ldots, x[5]) &= \frac{\alpha_1}{\alpha} \cdot \frac{\alpha_0}{\alpha+1} \cdot \frac{\alpha_0+1}{\alpha+2} \cdot \frac{\alpha_1+1}{\alpha+3} \cdot \frac{\alpha_1+2}{\alpha+4} \\ &= \frac{[\alpha_1(\alpha_1+1)(\alpha_1+2)][\alpha_0(\alpha_0+1)]}{\alpha \cdots (\alpha+4)}. \end{aligned}$$

$$\mathcal{D} = \langle H, T, T, H, H \rangle$$

ML score: $$\left(\frac{3}{5}\right)^3 \cdot \left(\frac{2}{5}\right)^2 = \frac{108}{3125} \approx 0.035.$$ >> Marginal likelihood $$\frac{[1 \cdot 2 \cdot 3] \cdot [1 \cdot 2]}{2 \cdot 3 \cdot 4 \cdot 5 \cdot 6} = \frac{12}{720} = 0.017$$

for $\alpha_1 = \alpha_0 = 1$

# Bayesian score

- Marginal likelihood for a single variable (easy to derive)

$$P(x[1],\ldots,x[M]) = \frac{\Gamma(\alpha)}{\Gamma(\alpha+M)} \cdot \prod_{i=1}^{k} \frac{\Gamma(\alpha_i + M[x^i])}{\Gamma(\alpha_i)}.$$

where $\Gamma$ is the *Gamma function* such that*:*

$$\Gamma(m) = (m-1)! \quad \text{and} \quad \Gamma(x+1) = x \cdot \Gamma(x)$$

- General Bayesian networks: marginal likelihood of Dirichlet priors

$$P(\mathcal{D} \mid \mathcal{G}) = \prod_i \prod_{\boldsymbol{u}_i \in Val(\mathrm{Pa}_{X_i}^{\mathcal{G}})} \frac{\Gamma(\alpha_{X_i|\boldsymbol{u}_i}^{\mathcal{G}})}{\Gamma(\alpha_{X_i|\boldsymbol{u}_i}^{\mathcal{G}} + M[\boldsymbol{u}_i])} \prod_{x_i^j \in Val(X_i)} \left[ \frac{\Gamma(\alpha_{x_i^j|\boldsymbol{u}_i}^{\mathcal{G}} + M[x_i^j, \boldsymbol{u}_i])}{\Gamma(\alpha_{x_i^j|\boldsymbol{u}_i}^{\mathcal{G}})} \right]$$

# Bayesian information score (BIC)

- BIC is an approximation for Bayesian score under certain conditions.

$$\text{score}_{BIC}(\mathcal{G} \ : \ \mathcal{D}) = \ell(\hat{\boldsymbol{\theta}}_{\mathcal{G}} : \mathcal{D}) - \frac{\log M}{2}\text{Dim}[\mathcal{G}].$$

$$\text{score}_{BIC}(\mathcal{G} \ : \ \mathcal{D}) = M\sum_{i=1}^{n}\boldsymbol{I}_{\hat{P}}(X_i; \text{Pa}_{X_i}) - M\sum_{i=1}^{n}\boldsymbol{H}_{\hat{P}}(X_i) - \frac{\log M}{2}\text{Dim}[\mathcal{G}]$$

- It provides a trade-off between model complexity and the likelihood.

# The BIC score is consistent

- We say a scoring function is consistent if the following properties hold as $M \rightarrow \infty$ with probability that approaches 1
  - The structure $g^*$ will maximize the score
  - All structures $g$ that are not I-equivalent to $g^*$ will have strictly lower score.

# Structure and parameter priors

- We need to specify actual choice of priors for Bayesian score

$$\text{score}_B(\mathcal{G} : \mathcal{D}) = \log P(\mathcal{D} \mid \mathcal{G}) + \log P(\mathcal{G}).$$

- Structure priors
  - We often use a uniform prior over structures $P(g) \propto c$.
  - We might penalize edges in the graph, $P(g) \propto c^{|g|}$ where $c < 1$ and $|g|$ is the number of edges in the graph.
  - Normalizing constant is the same across structures and can be ignored.

- Parameter priors
  - K2 prior: efficient to use! Bayesian score with this prior is not score equivalent!
  - Bayesian score with BDe prior satisfies score equivalence.

- **Score equivalence:** a scoring rule satisfies score equivalence if all I-equivalent networks have the same score for all data sets.
  - The likelihood score and the BIC score also satisfy score equivalence.

University of TEHRAN

# Structure search

# Optimization problem

- Input
  - Training set $D$
  - Scoring function (including priors, if needed)
  - A set of possible network structures
- Desired output: a network structure that maximizes the score
- Search algorithms will in general apply unchanged to different scores.
- Two important properties that can affect search
  - Decomposability

$$\mathrm{score}(\mathcal{G} : \mathcal{D}) = \sum_i \mathrm{FamScore}(X_i \mid \mathrm{Pa}^{\mathcal{G}}_{X_i} : \mathcal{D}),$$

  - Score equivalence

# Learning tree-structured networks

- Each variable in a tree-structured network has at most one parent.
- The notion of tree-structured networks also covers graphs composed of a set of disconnected trees (forest).

- Why do we care about trees?
  - Can be learned efficiently – in polynomial time.
  - Spare parameterization – avoid most overfitting problems
  - Often used as a starting point for learning a more complex structure.

# Learning tree-structured networks

- We will try to maximize the difference between the score of a tree structure $g$ and the score of the empty structure $g_0$

$$\Delta(\mathcal{G}) = \text{score}(\mathcal{G} : \mathcal{D}) - \text{score}(\mathcal{G}_\emptyset : \mathcal{D})$$

we can easily obtain

$$\Delta(\mathcal{G}) = \sum_{i, \mathrm{Pa}^{\mathcal{G}}_{X_i} \neq \emptyset} \left( \text{FamScore}(X_i \mid \mathrm{Pa}^{\mathcal{G}}_{X_i} : \mathcal{D}) - \text{FamScore}(X_i : \mathcal{D}) \right).$$

we define the weights

$$w_{j \to i} = \text{FamScore}(X_i \mid X_j : \mathcal{D}) - \text{FamScore}(X_i : \mathcal{D}),$$

simply we have

$$\Delta(\mathcal{G}) = \sum_{X_j \to X_i \in \mathcal{G}} w_{j \to i}.$$

# Learning tree-structured networks

- if the score satisfies score equivalence, we have $w_{i \to j} = w_{j \to i}$

- **Algorithm (if the score satisfies score equivalence):**
  - Define a weighted complete undirected graph with $w_{i \to j}$.
  - Find a maximum undirected spanning tree.
  - Remove all edges with weight zero to produce a forest.
  - Choose an arbitrary node and direct all edges away

# Learning general Bayesian networks

- **Theorem:** Finding the maximum-score network with at most $d \geq 2$ parents for each variable is *NP-hard*.

- We will use heuristic algorithm to search the space of graphs and return a high-scoring one.

- Local operators:
  - Edge addition
  - Edge deletion
  - Edge reversal

- Search techniques
  - Greedy hill climbing
  - Simulated annealing
  - …