Structure Learning

Learning

Professor Ajoodha

Lecture 8

School of Computer Science and Applied Mathematics
The University of the Witwatersrand, Johannesburg

ExplainableAI Lab

— MODELLING. DECISION MAKING. CAUSALITY —

Problem Statement

- We assume that $\mathcal{D} = \{\xi[1], \ldots, \xi[M]\}$ is **generated IID** from $P^*(\mathcal{X})$.
- $P^*(\mathcal{X})$ **is induced** by a Bayesian network $\mathcal{G}^*$ over $\mathcal{X}$.
- To what extent do the independencies in $\mathcal{G}^*$ **manifest** in $\mathcal{D}$

Problem

Find the Bayesian network structure, $\mathcal{G}$, that best represents the dependencies which manifest in $\mathcal{D}$.

Why would we want to do this?

- The importance of the reconstruction depends on the learning goal:
    - **Knowledge Discovery**: learn the dependency structure relating variables in our domain.
    - **Density estimation**: to estimate a statistical model of the underlying distribution.

Knowledge Discovery

- If the task is for **knowledge discovery** then we need to reconstruct $\mathcal{G}^*$.
- However there are many perfect maps for $\mathcal{G}^*$ in its I-equavalence class.
- $\mathcal{G}^*$ is **not identifiable** from $\mathcal{D}$
- Data sampled from P^* is noisy
- When learning $\mathcal{G}^*$:
  1. If we learn too **many edges**, then we learn deceptive edges
  2. If we learn too **few edges**, then we cannot capture the true distribution

Density Estimation

- Process of estimating the **probability density function** of a random variable from a set of observations.
- In this case, we want the learned model to **generalise** to new instances.
- $\mathcal{G}^*$ is ideal for this task.
- **More edges are better** than too few if $\mathcal{G}^*$ is unknown, as a complex structure can still capture P^*.
- However, in the case of limited data, sparser structures generalise better.
- That is, simple structures can **improve generalization** despite inability to represent true distribution.

Overview of Methods

- There are three approaches to structure learning:
  1. **Constraint-based approaches**:
     - Bayesian network is a representation of independencies
     - Test for dependence/independence in the data
     - Find I-map/P-map that best explain dependence/independence
  2. **Score-based approaches**:
     - Bayesian network specifies a statistical model: model selection problem
     - Define a hypothesis space of potential models
     - Define scoring function: calculates model fit to data
     - Search for highest-scoring network structure using scoring function
  3. **Bayesian model averaging**:
     - Generates an ensemble of possible structures
     - Tries to average the prediction of all possible structures
     - Sometimes can be done efficiently

Constraint-based Structure Learning

- We aim to capture the network structure that accurately represents the **independencies in the domain**.
- The basic idea is to build the best minimal I-map
- How can we answer independence queries?

  e.g. Does P $\models (X_i \perp\!\!\!\perp \{X_1, \ldots, X_{i-1}\} - \mathbf{U} \mid \mathbf{U})$?
- Recall the `Build-Minimal-I-Map`.
  - To determine a parent of X_i it must examine all 2^{i-1} possible subsets of $X_1, \ldots, X_{i-1}$
- We can do better by making some assumptions:
  1. bounding the indegree: $|Pa_{X_i}^{\mathcal{G}^*}| \leq d$
  2. Independence queries can answer queries up to $2d + 2$ variables
  3. P^* is faithful to $\mathcal{G}^*$

Hypothesis Testing

- Hypothesis testing: $X \perp Y$
- Null Hypothesis (H_0): $P^*(X,Y) = \hat{P}(X)\hat{P}(Y)$
- Empirical Mutual information, $\mathbb{I}_{\hat{P}_{\mathcal{D}}}(X;Y)$ is often used:

$$\boxed{d_{\mathbb{I}}(\mathcal{D}) = \mathbb{I}_{\hat{P}_{\mathcal{D}}}(X;Y) = \sum_{x,y} \frac{M[x,y]}{M} \log \frac{M[x,y]/M}{M[x]/M \cdot M[y]/M}}$$

$$\mathtt{R}_{d,t}(\mathcal{D}) = \begin{cases} \mathtt{Accept} & \text{if } d(\mathcal{D}) \leq t \\ \mathtt{Reject} & \text{if } d(\mathcal{D}) > t \end{cases}$$

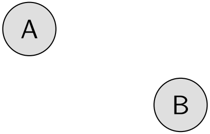
- **Intuition**: Accepts hypothesis if deviance is small and rejects if deviance is large.

$$\text{p-value}(t) = P(\{\mathcal{D} : d(\mathcal{D}) > t\} \mid H_0, M)$$

Example

1. p-value(t) $= 0.01$
2. Ordering: A, B, C.
3. **Add** A and B

Learned Model:



$\mathcal{D}$	$\langle A, B, C\rangle$
$\xi[1]$	$\langle a^1, b^0, c^1\rangle$
$\xi[2]$	$\langle a^0, b^1, c^0\rangle$
$\xi[3]$	$\langle a^0, b^0, c^1\rangle$
$\xi[4]$	$\langle a^1, b^1, c^0\rangle$
$\xi[5]$	$\langle a^0, b^0, c^1\rangle$

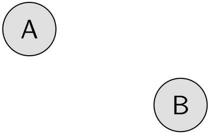
$$\mathbb{I}_{\hat{P}_{\mathcal{D}}}(X;Y)$$

$$= \sum_{x,y} \frac{M[x,y]}{M} \log \frac{M[x,y]/M}{M[x]/M \cdot M[y]/M}$$

Example

1. p-value(t) $= 0.01$
2. Ordering: A, B, C.
3. **Add** A and **Test** $A \perp B$

Learned Model:



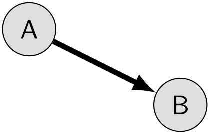
$\mathcal{D}$	$\langle A, B, C \rangle$
$\xi[1]$	$\langle a^1, b^0, c^1 \rangle$
$\xi[2]$	$\langle a^0, b^1, c^0 \rangle$
$\xi[3]$	$\langle a^0, b^0, c^1 \rangle$
$\xi[4]$	$\langle a^1, b^1, c^0 \rangle$
$\xi[5]$	$\langle a^0, b^0, c^1 \rangle$

$$\begin{aligned} &= \frac{M[a_0, b_0]}{M} \log \frac{M[a_0, b_0]/M}{M[a_0]/M \cdot M[b_0]/M} \\ &+ \frac{M[a_0, b_1]}{M} \log \frac{M[a_0, b_1]/M}{M[a_0]/M \cdot M[b_1]/M} \\ &+ \frac{M[a_1, b_0]}{M} \log \frac{M[a_1, b_0]/M}{M[a_1]/M \cdot M[b_0]/M} \\ &+ \frac{M[a_1, b_1]}{M} \log \frac{M[a_1, b_1]/M}{M[a_1]/M \cdot M[b_1]/M} \end{aligned}$$

Example

1. p-value(t) $= 0.01$
2. Ordering: A, B, C.
3. **Add** A and **Test** $A \perp B$

Learned Model:



$\mathcal{D}$	$\langle A, B, C \rangle$
$\xi[1]$	$\langle a^1, b^0, c^1 \rangle$
$\xi[2]$	$\langle a^0, b^1, c^0 \rangle$
$\xi[3]$	$\langle a^0, b^0, c^1 \rangle$
$\xi[4]$	$\langle a^1, b^1, c^0 \rangle$
$\xi[5]$	$\langle a^0, b^0, c^1 \rangle$

$$= \frac{2}{5} \log \frac{2/5}{3/5 \cdot 3/5} + \frac{1}{5} \log \frac{1/5}{3/5 \cdot 2/5}$$

$$+ \frac{1}{5} \log \frac{1/5}{2/5 \cdot 3/5} + \frac{1}{5} \log \frac{1/5}{2/5 \cdot 2/5}$$

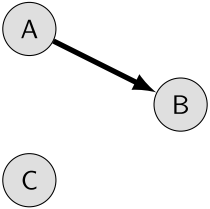
$$= 0.042 - 0.036 - 0.036 + 0.044$$

$$= 0.014 > 0.01$$

Example

1. p-value(t) $= 0.01$
2. Ordering: A, B, C.
3. **Add** C and **Test** $A \perp C$

Learned Model:



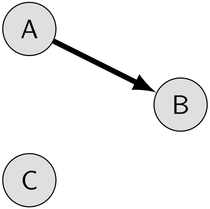
$\mathcal{D}$	$\langle A, B, C \rangle$
$\xi[1]$	$\langle a^1, b^0, c^1 \rangle$
$\xi[2]$	$\langle a^0, b^1, c^0 \rangle$
$\xi[3]$	$\langle a^0, b^0, c^1 \rangle$
$\xi[4]$	$\langle a^1, b^1, c^0 \rangle$
$\xi[5]$	$\langle a^0, b^0, c^1 \rangle$

$$
\begin{aligned}
&= \frac{M[a_0, c_0]}{M} \log \frac{M[a_0, c_0]/M}{M[a_0]/M \cdot M[c_0]/M} \\
&+ \frac{M[a_0, c_1]}{M} \log \frac{M[a_0, c_1]/M}{M[a_0]/M \cdot M[c_1]/M} \\
&+ \frac{M[a_1, c_0]}{M} \log \frac{M[a_1, c_0]/M}{M[a_1]/M \cdot M[c_0]/M} \\
&+ \frac{M[a_1, c_1]}{M} \log \frac{M[a_1, c_1]/M}{M[a_1]/M \cdot M[c_1]/M}
\end{aligned}
$$

Example

1. p-value(t) $= 0.01$
2. Ordering: A, B, C.
3. **Add** C and **Test** $A \perp C$

Learned Model:



$\mathcal{D}$	$\langle A, B, C \rangle$
$\xi[1]$	$\langle a^1, b^0, c^1 \rangle$
$\xi[2]$	$\langle a^0, b^1, c^0 \rangle$
$\xi[3]$	$\langle a^0, b^0, c^1 \rangle$
$\xi[4]$	$\langle a^1, b^1, c^0 \rangle$
$\xi[5]$	$\langle a^0, b^0, c^1 \rangle$

$$= \frac{1}{5} \log \frac{1/5}{3/5 \cdot 2/5} + \frac{2}{5} \log \frac{2/5}{3/5 \cdot 3/5}$$

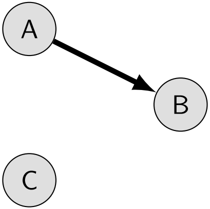
$$+ \frac{1}{5} \log \frac{1/5}{2/5 \cdot 2/5} + \frac{1}{5} \log \frac{1/5}{2/5 \cdot 3/5}$$

$$= -0.02 + 0.02 + 0.02 - 0.02 = 0 < 0.01$$

Example

1. p-value(t) $= 0.01$
2. Ordering: A, B, C.
3. **Test** $C \perp B$

Learned Model:



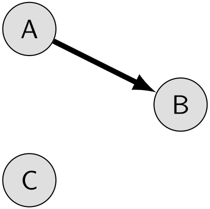
$\mathcal{D}$	$\langle A, B, C\rangle$
$\xi[1]$	$\langle a^1, b^0, c^1\rangle$
$\xi[2]$	$\langle a^0, b^1, c^0\rangle$
$\xi[3]$	$\langle a^0, b^0, c^1\rangle$
$\xi[4]$	$\langle a^1, b^1, c^0\rangle$
$\xi[5]$	$\langle a^0, b^0, c^1\rangle$

$$
\begin{aligned}
&= \frac{M[b_0, c_0]}{M} \log \frac{M[b_0, c_0]/M}{M[b_0]/M \cdot M[c_0]/M} \\
&+ \frac{M[b_0, c_1]}{M} \log \frac{M[b_0, c_1]/M}{M[b_0]/M \cdot M[c_1]/M} \\
&+ \frac{M[b_1, c_0]}{M} \log \frac{M[b_1, c_0]/M}{M[b_1]/M \cdot M[c_0]/M} \\
&+ \frac{M[b_1, c_1]}{M} \log \frac{M[b_1, c_1]/M}{M[b_1]/M \cdot M[c_1]/M}
\end{aligned}
$$

Example

1. p-value(t) $= 0.01$
2. Ordering: A, B, C.
3. **Test** $C \perp B$

Learned Model:



$\mathcal{D}$	$\langle A, B, C \rangle$
$\xi[1]$	$\langle a^1, b^0, c^1 \rangle$
$\xi[2]$	$\langle a^0, b^1, c^0 \rangle$
$\xi[3]$	$\langle a^0, b^0, c^1 \rangle$
$\xi[4]$	$\langle a^1, b^1, c^0 \rangle$
$\xi[5]$	$\langle a^0, b^0, c^1 \rangle$

$$= \frac{0}{5} \log \frac{0/5}{3/5 \cdot 2/5} + \frac{3}{5} \log \frac{3/5}{3/5 \cdot 3/5}$$

$$+ \frac{2}{5} \log \frac{2/5}{2/5 \cdot 2/5} + \frac{0}{5} \log \frac{0/5}{2/5 \cdot 3/5}$$

$$= 0 - 0.18 - 0.12 + 0$$

$$= -0.3 < 0.01$$

Limitations

- We can evaluate independence queries in the `Build-Minimal-I-Map` procedure.
- When the test rejects the null hypothesis we treat the variables **as dependent**.
- With an error of 95%, we get 1 in 20 rejections wrong.
- Multiple hypothesis testing can harm network reconstruction accuracy by **increasing incorrect conclusions**.
- Errors in independence tests can lead to multiple errors in the PDAG constructed by `Build-Minimal-I-Map`.
- In practice works well with **few variables and large sample** sizes

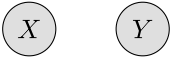
Score-based Structure Learning Approaches

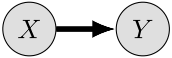
- Approaches problem using optimisation:
  1. **Score** each candidate structure using score function
  2. **Search** for both a graph $\mathcal{G}$ and parameters set $\boldsymbol{\theta}_{\mathcal{G}}$ that makes the data as probable as possible.
- This can be done by defining $\mathcal{M} = \langle \mathcal{G}, \boldsymbol{\theta}_{\mathcal{G}} \rangle$ using the maximum likelihood parameters: $\hat{\boldsymbol{\theta}}_{\mathcal{G}}$

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

- **Intuition**: To maximise likelihood $(\mathcal{G}; \theta_{\mathcal{G}})$, we determine $\mathcal{G}$ that produces the highest likelihood to $\mathcal{D}$.

Decomposition of the Likelihood

$\mathcal{G}_0$:


$\mathcal{G}_1$:


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

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

$$\begin{aligned}
&\text{score}_L(\mathcal{G}_1 : \mathcal{D}) - \text{score}_L(\mathcal{G}_0 : \mathcal{D}) \\
&= \sum_m \left(\log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]|x[m]}\right) - \sum_m \left(\log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]}\right) \\
&= \sum_m \left(\log \hat{\theta}_{y[m]|x[m]} - \log \hat{\theta}_{y[m]}\right) \\
&= \sum_{x,y} \left(M[x,y] \log \hat{\theta}_{y|x}\right) - \sum_y \left(M[y] \log \hat{\theta}_y\right)
\end{aligned}$$

Decomposition of the Likelihood

$$
\begin{aligned}
&= \sum_{x,y}\Big(M[x,y]\log\frac{M[x,y]}{M[x]}\Big) - \sum_{y}\Big(M[y]\log\frac{M[y]}{M}\Big) \\
&= M\sum_{x,y}\Big(\hat{P}(x,y)\log\hat{P}(y \mid x)\Big) - M\sum_{y}\Big(\hat{P}(y)\log\hat{P}(y)\Big) \\
&= M\Big(\sum_{x,y}\hat{P}(x,y)\log\hat{P}(y \mid x) - \sum_{x,y}\hat{P}(x,y)\log\hat{P}(y)\Big) \\
&= M\sum_{x,y}\hat{P}(x,y)\log\frac{\hat{P}(x,y)}{\hat{P}(x)\hat{P}(y)} \\
&= M\ \mathbb{I}_{\hat{P}_{\mathcal{D}}}(X;Y)
\end{aligned}
$$

Intuition: $\mathbb{I}_{\hat{P}_{\mathcal{D}}}(X;Y)$ is the averaged distance between the joint distribution, $\hat{P}(X,Y)$, and the product of marginals, $\hat{P}(X)\hat{P}(Y)$, for X and Y.

Generalisation of Likelihood Score

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

$$\mathbb{I}_{\hat{P}_{\mathcal{D}}}(X_i; Pa_{X_i}^{\mathcal{G}}) = \sum_{\mathbf{u}_i} \sum_{\mathbf{x}_i} \hat{P}(x_i, \mathbf{u}_i) \log \frac{\hat{P}(x_i, \mathbf{u}_i)}{\hat{P}(x_i)\hat{P}(\mathbf{u}_i)}$$

$$\mathbb{H}_{\hat{P}_{\mathcal{D}}}(X_i) = \sum_{x_i} \hat{P}(x_i) \log \frac{1}{\hat{P}(x_i)}$$

Intuition: The likelihood score measure the strength of the dependencies between the variables and their parents.

Limitations of Likelihood Score

- **Computationally** expensive as the number of variables and instances increase.
- Favours **over-fitting** models relative to the training data.
- Learns the empirical distribution, which may not necessarily be the true distribution P^*, thus it may not accurately reflect the true likelihood of the data.
- Different models may have the **same likelihood**, which makes it difficult to identify the true underlying model.
- Likelihood score will always "prefer" a **more complicated structure** given random noise in the data.

The Bayesian Paradigm

$$\overbrace{P(\mathcal{G} \mid \mathcal{D})}^{\texttt{posterior}} = \frac{\overbrace{P(\mathcal{D} \mid \mathcal{G})}^{\texttt{likelihood}}\,\overbrace{P(\mathcal{G})}^{\texttt{prior}}}{\underbrace{P(\mathcal{D})}_{\texttt{constant}}}$$

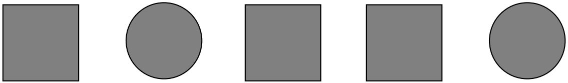
$$\text{score}_B(\mathcal{G} : \mathcal{D}) = \log P(\mathcal{D} \mid \mathcal{G}) + \overbrace{\log P(\mathcal{G})}^{\texttt{structure preference}}$$

$$P(\mathcal{D} \mid \mathcal{G}) = \int_{\Theta_{\mathcal{G}}} \overbrace{P(\mathcal{D} \mid \boldsymbol{\theta}_{\mathcal{G}}, \mathcal{G})}^{\texttt{marginal likelihood}} \overbrace{P(\boldsymbol{\theta}_{\mathcal{G}} \mid \mathcal{G})}^{\texttt{parameter prior}} \, d\boldsymbol{\theta}_{\mathcal{G}}$$

Marginal Likelihood VS Maximum Likelihood

- Both calculate on $L(\mathcal{D} : \langle \boldsymbol{\theta}_{\mathcal{G}}, \mathcal{G} \rangle)$, but ...
- **Maximum likelihood** calculates the *maximum*
- **Marginal Likelihood** calculates the *average*, w.r.t. $P(\boldsymbol{\theta}_{\mathcal{G}} \mid \mathcal{G})$
- The Bayesian approach models that $\boldsymbol{\theta}_{\mathcal{G}}$ is **not the only choice** of parameters given the training set $\mathcal{D}$.
- Integrating $P(\mathcal{D} \mid \boldsymbol{\theta}_{\mathcal{G}}, \mathcal{G})$ over possibilities of parameters allows us to measure the **expected likelihood**.

Marginal Likelihood VS Maximum Likelihood



1. Calculate the maximum likelihood of the data (frequency of shapes).

   Solution:

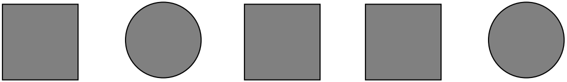
$$P(\mathcal{D} \mid \hat{\theta}) = \left(\frac{M[S]}{M}\right)^{M[S]} \cdot \left(\frac{M[C]}{M}\right)^{M[C]} = \left(\frac{3}{5}\right)^{3} \cdot \left(\frac{2}{5}\right)^{2} = \frac{108}{3125} \approx 0.035$$

2. Now, calculate the marginal likelihood of the data with the prior $\mathrm{Dir}(\alpha_0, \alpha_1)$ of the frequency of shapes.

   Solution: One approach can use the integral as before, or we can use the chain rule of probabilities:

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

Marginal Likelihood



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

where $M^m[1]$ is the number of squares in m examples.

$$P(x[1], \ldots, x[5]) = \frac{\alpha_0}{\alpha} \cdot \frac{\alpha_1}{\alpha+1} \cdot \frac{\alpha_0+1}{\alpha+2} \cdot \frac{\alpha_0+2}{\alpha+3} \cdot \frac{\alpha_1+1}{\alpha+4}$$

$$= \frac{[(\alpha_0)(\alpha_0+1)(\alpha_0+2)][(\alpha_1)(\alpha_1+1)]}{\alpha \cdot (\alpha+1) \cdots (\alpha+4)}$$

- Picking $\alpha_0 = \alpha_1 = 1$, so that $\alpha = \alpha_0 + \alpha_1 = 2$, then

$$P(x[1], \ldots, x[5]) = \frac{[(1)(2)(3)][(1)(2)]}{2 \cdot 3 \cdot 4 \cdot 5 \cdot 6} = \frac{12}{720} \approx 0.017$$

Bayesian Score

- A maximum likelihood model assigns **higher probability** to a sequence than marginal likelihood.
- Log-likelihood is **over-optimistic** as it uses hindsight-optimized parameter for entire sequence fit.
- The general binomial distribution with Beta prior is:

$$P(x[1], \ldots, x[M]) = \frac{[\alpha_1 \cdots (\alpha_1 + M[1] - 1)][\alpha_0 \cdots (\alpha_0 + M[0] - 1)]}{\alpha \cdots (\alpha + M - 1)}$$

- Since $\Gamma(m) = (m-1)!$ and $\Gamma(x+1) = x\Gamma(x)!$, then

$$P(x[1], \ldots, x[M]) = \frac{\frac{\Gamma(\alpha_1 + M[1])}{\Gamma(\alpha)} \cdot \frac{\Gamma(\alpha_0 + M[0])}{\Gamma(\alpha_0)}}{\frac{\Gamma(\alpha)}{\Gamma(\alpha + M)}}$$

Marginal Likelihood for Multinomial Distributions

- This very comfortably extends for the multinomial as:

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

- Note that we use the same sufficient statistics to compute the marginal likelihood as we do the maximum likelihood

The Bayesian Score

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

$$\prod_i \prod_{\mathbf{u}_i \in Val(Pa^{\mathcal{G}}_{X_i})} \frac{\Gamma(\alpha^{\mathcal{G}}_{X_i|\mathbf{u}_i})}{\Gamma(\alpha^{\mathcal{G}}_{X_i|\mathbf{u}_i} + M[\mathbf{u}_i])} \prod_{\mathbf{x}_i^j \in Val(X_i)} \left[\frac{\Gamma(\alpha^{\mathcal{G}}_{x_i^j|\mathbf{u}_i} + M[x_i^j, \mathbf{u}_i])}{\Gamma(\alpha^{\mathcal{G}}_{x_i^j|\mathbf{u}_i})} \right]$$

- where $\alpha^{\mathcal{G}}_{X_i|\mathbf{u}_i} = \sum_j \alpha^{\mathcal{G}}_{x_i^j|\mathbf{u}_i}$
- In practice we use the logarithm for manageable computation.
- The Bayesian score is biased to simple structures, but as it gets more data it “recognises” that a more complex structure is necessary
- It trades off fit to data with model complexity, thereby **reducing** the extent of overfitting

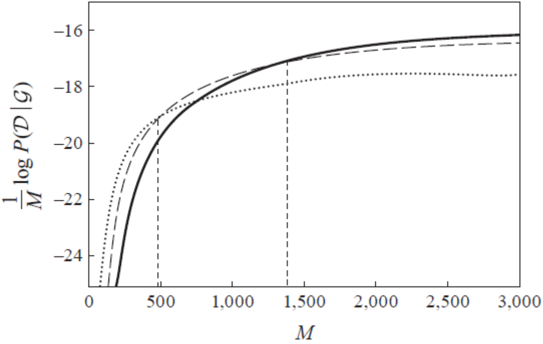
The BIC Score

If we use a Dirichlet prior for all parameters, then as $M \to \infty$, we have:

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

- Negation leads to **Minimum Description Length**: gives bits needed to encode model and data, given the model.
- Mutual information term **grows linearly** in M; whereas the complexity term **grows logarithmically**.
- The larger M is, the more emphasis will be **given to the fit** to data

Trade off: Fit and Complexity



- **Solid**: Original structure (509 parameters)
- **Dashed**: Simplification (359 parameters)
- **Dotted**: Tree-structure (214 parameters)

Credit: *Koller Textbook [2009], pp 802*

Structure Priors

$$\text{score}_B(\mathcal{G} : \mathcal{D}) = \log P(\mathcal{D} \mid \mathcal{G}) + \overbrace{\log P(\mathcal{G})}^{\texttt{structure preference}}$$

- Marginal likelihood grows linearly, but the structure prior **is constant** (minor influence), but matters for small samples!
- E.g. $P(\mathcal{G}) \propto c^{|\mathcal{G}|}$, $c < 1$ & $|\mathcal{G}|$ is edge count.
- It is useful for prior to satisfy **structural modularity**:
- That is, the prior for each term relates to the prior for that family

$$P(\mathcal{G}) \propto \prod_i P(Pa_{X_i} = Pa^{\mathcal{G}}_{X_i})$$

- It is also useful to make I-equivalent network structures have **the same prior.**

Parameter Priors

$$P(\mathcal{D} \mid \mathcal{G}) = \int_{\Theta_{\mathcal{G}}} \overbrace{P(\mathcal{D} \mid \boldsymbol{\theta}_{\mathcal{G}}, \mathcal{G})}^{\texttt{marginal likelihood}} \quad \overbrace{P(\boldsymbol{\theta}_{\mathcal{G}} \mid \mathcal{G})}^{\texttt{parameter prior}} \quad d\boldsymbol{\theta}_{\mathcal{G}}$$

- How can we represent parameter priors since the number of structures are super exponential?
- **K2 Prior:** A fixed Dirichlet distribution, e.g. $\alpha = 1$, for every parameter. But, this double counts priors for different structures.
- **BDe Prior (Bayesian Dirichlet equivalence):** Elicits a prior probability distribution P' over the entire probability space.

$$\alpha_{x_i|pa_{X_i}} = \alpha \cdot P'(x_i, pa_{X_i})$$

Important Properties

- The BIC and Bayesian scores satisfy **consistency**:
  1. $\mathcal{G}^*$ will maximise the score.
  2. Structures that are not I-equivalent to $\mathcal{G}^*$ will have *strictly lower* score
- The BIC and likelihood scores satisfy **decomposability**:
  1. decomposable if can be written as a sum of family scores:

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

  2. Structure search procedures take advantage of score decomposability to save computational time and space.
- Likelihood, BIC, BDe scores satisfy **score equivalence**:
  1. Networks from the same equivalence class have the same score.

Structure Search

- We now have a well-defined optimisation problem:

  Input:

  1. A training set $\mathcal{D}$
  2. scoring function (including priors)
  3. A set of network structures $\mathcal{G}$ (incorporating priors)

  Output:

  1. A set of i-equivalent structures that maximizes the score with resect to the data

Why Learn a Tree

- There are many advantages to learning trees:
  1. Trees can be learned efficiently (polynomial time)
  2. Sparse and avoid overfitting
  3. Capture most important dependencies (domain insight)
  4. Baseline for approximating distribution
  5. Used as a prior for graph search

Learning Tree Structured Networks

- To complete the tree:
  1. If the score is decomposable then:

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

  2. If score equivalent then $w_{i\to j} = w_{j\to i}$
  3. Calculate the maximum weighted spanning forest in a directed weighted graph.
  4. Complexity: $O(\overbrace{n^2 \cdot M}^{\text{Sufficient Statistics}} + \overbrace{n^2 \log n}^{\text{Learning MWSF}})$
- For the likelihood score, $w_{i\to j}$ will be positive (tree).
- For the BIC and BDe score, $w_{i\to j}$ can be negative (forest).

Learning General Graphs

- The following is $\mathcal{NP}$-hard for any $d \geq 2$:

$$\mathcal{G}^* = \operatorname*{argmax}_{\mathcal{G} \in \boldsymbol{\mathcal{G}}} \mathsf{score}(\mathcal{G} : \mathcal{D})$$

- As with many intractable problems, we resort to heuristic combinatorial optimisation method:
  - A search space
  - A scoring function
  - A search procedure

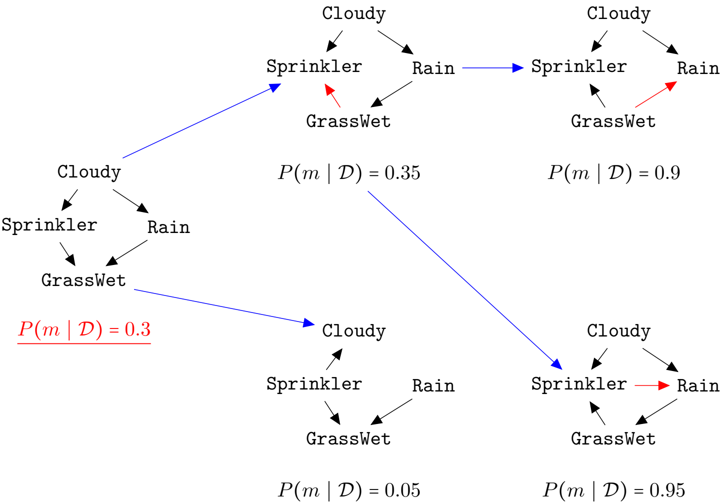
Search Space

- Search space = graph of solutions linked by operators.
  1. Edge addition
  2. Edge deletion
  3. Edge reversal
- Properties of search space:
  1. Diameter of search space is n^2
  2. Local operators means local change in the score

The Search Procedure: Greedy Hill Climbing

- Now that we have a search space, we need a way to explore it.
- There are many local heuristic techniques, we look at **Greedy hill climbing**
  1. Pick a starting point $\mathcal{G}^t$ (random, prior, tree)
  2. Compute the score
  3. List the neighbouring structures $\mathcal{G}^t$: $\{\mathcal{G}_0^t, \ldots \mathcal{G}_S^t\}$
  4. Compute: $\{\text{score}(\mathcal{G}_0^t), \ldots \text{score}(\mathcal{G}_S^t)\}$
  5. Move to $\mathcal{G}^t = \max_{\mathcal{G}}\{\text{score}(\mathcal{G}_1^t), \ldots \text{score}(\mathcal{G}_S^t)\}$
  6. Repeat 2 to 5 until no modification improves the score
- **Intuition**: We make the change that **best improves** the score until no improve can be made.

Example


Cloudy
Sprinkler
Rain
GrassWet
$P(m \mid \mathcal{D}) = 0.3$
Cloudy
Sprinkler
Rain
GrassWet
$P(m \mid \mathcal{D}) = 0.35$
Cloudy
Sprinkler
Rain
GrassWet
$P(m \mid \mathcal{D}) = 0.9$
Cloudy
Sprinkler
Rain
GrassWet
$P(m \mid \mathcal{D}) = 0.05$
Cloudy
Sprinkler
Rain
GrassWet
$P(m \mid \mathcal{D}) = 0.95$

Computational Complexity

We need to evaluate the computational complexity:

1. It takes $O(n^2)$ to score the initial network
2. With K steps to convergence, and a decomposible score, it takes $O(K \cdot n^2)$ operator applications
   - Number of Families: $O(n)$
   - Acyclity check: Topological sort $O(|E|)$
   - Collect Sufficient Statistics: $O(M)$
3. Total complexity: $O(n^2 + K(n^2)(Mn + |E|))$:

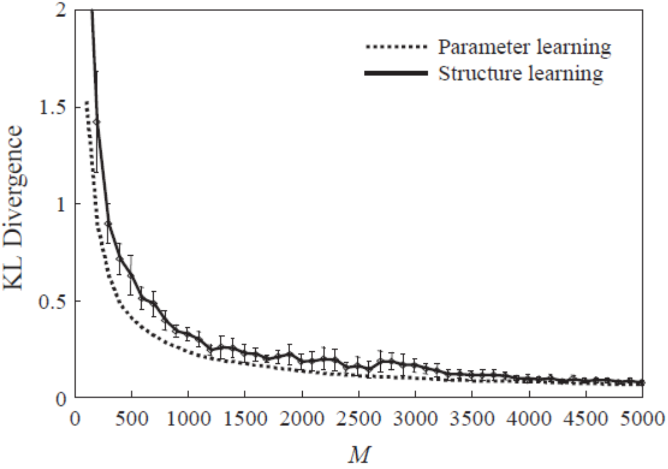
$$O(\underbrace{n^2}_{\text{Initial}} + K\overbrace{(n^2)}^{\text{Ops}}(\underbrace{Mn}_{\text{Sufficient Statistics per Family}} + \overbrace{|E|}^{Edges}))$$

4. Computational savings:
   - Keep score of operators in heap: update $O(n \log n)$; retrieve $O(1)$.

Local Maxima

- Once the procedure is complete, we could have two scenarios:
    1. **Local maximum:** All changes are score reducing.
    2. **Plateau:** neighbouring networks with same score
- Some approaches can assist with these problems:
    1. **Tabu search:** avoids considering recently applied operators by keeping a list of them.
    2. **Random Restarts:** Takes random steps in the search space.
    3. **Data Perturbation:** Randomly weighting data instances to overcome local obstacles
- **Simulated annealing**: Occasional moves to worse structures to explore a wider search space and escape local maxima
- Greedy Hill climbing with random restarts and tabu lists performs better than simulated annealing

Performance of Structure and Parameter Learning



Credit: *Koller Textbook [2009], pp 820*

Summary

- **Learning Goal**: density estimation or knowledge discovery
- **Approaches**: Constraint-based vs Score-based vs Bayesian
- **Scores**: likelihood, BIC, AIC, Bayesian, BDe
- **Likelihood**: Marginal likelihood vs maximum likelihood
- **Priors**: Structure and parameter
- **Structure Search**: Learning trees and graphs