

Previously...

Structure Learning

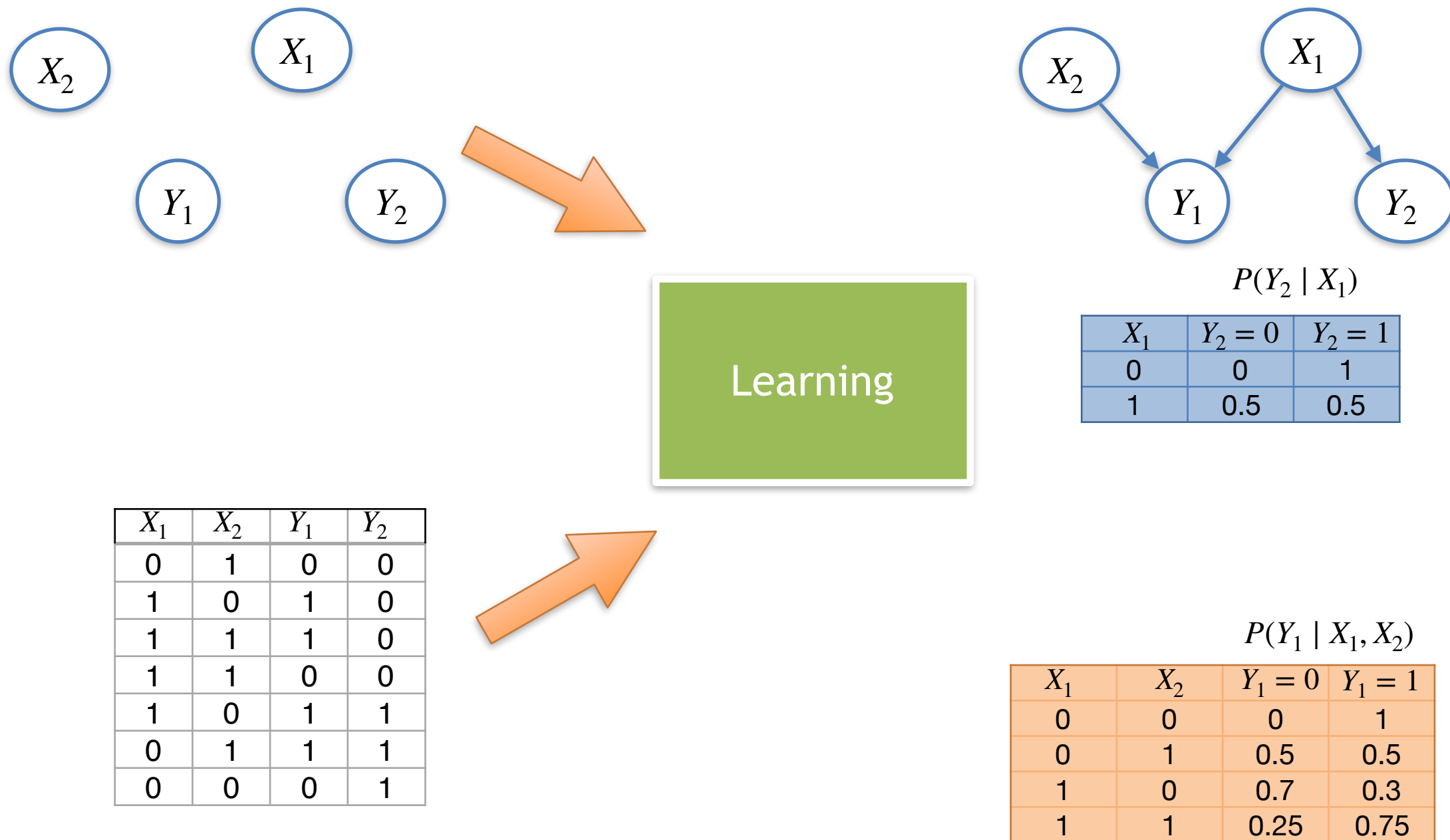
Lack of a priori independence knowledge

We assume we have a dataset, but don't know the independence assumptions we should make.

No missing data

For simplicity, we assume that the dataset is complete (there are no missing observations).

Unknown Structure and Complete Data



Assessing Empirical Independence

- Given a dataset of observations we can decide if two variables X and Y are independent using
 - the **mutual information (MI)** as the “difference” between $p(X, Y)$ and $p(X)p(Y)$

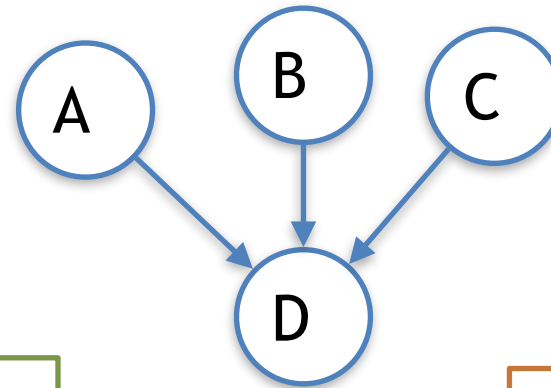
$$MI \equiv \sum_{x,y \in \text{dom}(X,Y)} P(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

where $p(X)$, $p(Y)$ and $p(X, Y)$ are the empirical distributions formed using the dataset

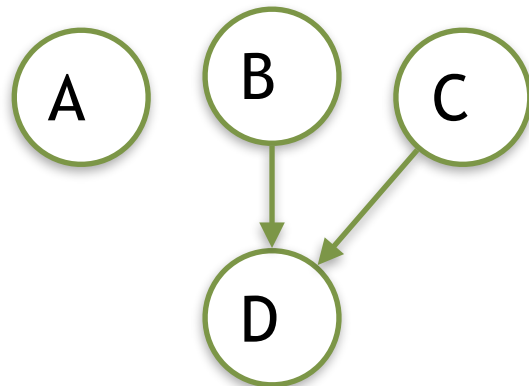
- MI is **equal to 0** if and only if X and Y are **independent**
- **Problem:** Since we formed $p(x)$ and $p(y)$ **based on a set of observations**, it's likely that, even for data sampled from a true distribution for which $x \perp y$, then the MI will not be zero.

Importance of Accurate Structure

True model

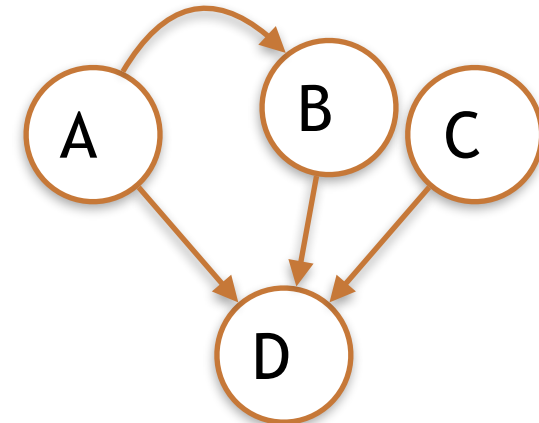


Missing Edge



- Incorrect Dependencies
- Correct distribution P^* cannot be learned
- Can generalise better

Spurious Edge



- Spurious dependencies
- Can correctly learn P^*
- Increases the number of parameters
- Worse generalisation

Learning BN structures (trees)

- The problem of learning a BN structure can be seen as an **optimisation problem** consisting of two parts:
 1. Define a scoring function
 2. Come up with an algorithm optimising the scoring function
- For trees we can apply a **greedy algorithm** for finding the maximum weight spanning tree.

SEARCHING OVER STRUCTURES (II)

Learning as an Optimisation Problem

Input:

- Training data
- Scoring function
- Set of possible structures

Output:

- A network that maximises the score

Greedy Hill Climbing

- **Initial G :**
 - ◉ Empty Network
 - ◉ Best tree
 - ◉ A random network
 - ◉ A network constructed using prior knowledge
- **At each iteration:**
 - Consider score for **all possible changes** (based on the operator we have defined)
 - Apply change that **maximally improve** the score
- **Stop when no modification improves the score**

Beyond Trees

Finding maximal scoring network structures with at most k parents for each variable is **NP-hard** for $k > 1$.

- ❖ Thus, using the simple greedy algorithm is not longer guaranteed to work e.g. on structures allowing two parents.
- ❖ No efficient algorithm is likely to be found for this problem

Heuristic Search

- To tackle this problem we can use **heuristic hill climbing** search

Given a BN G , we can consider **changes** to the network and see whether these changes **improve its score**.

For example: add/remove edges, reverse an edge

Design Choices

- **Search operators:**
 - ▶ Local step
 - ▶ Global steps
- **Search techniques:**
 - ▶ Greedy hill-climbing
 - ▶ Best first search
 - ▶ Simulated Annealing,
 - ▶ ...

??

Which of the following might pose problems for learning the correct structure?

1. Discrete steps in the score while changing the structure
2. Small changes of the network lead to no or very small changes in the score
3. Local maxima
4. The inability to express edge deletion as an atomic operation on the network structure

Pitfalls

- Greedy hill-climbing can get stuck in :
 - **Local maxima**
 - **Plateau:** changes on the network do not affect the score
 - ❖ **Equivalent networks** are often neighbours in the search space

Simple Algorithm to avoid Pitfalls

Greedy hill-climbing

+

► **Random restarts**

Take a number of random steps when stuck, and then start climbing again

► **Forbidden list**

- Keep a list of the last N steps taken
- Search cannot reverse any of the steps in the list

Summary

- BN structure learning is useful for building better predictive models
 - when domain experts don't know the structure well, and
 - for Knowledge discovery
- Finding highest-scoring structures is NP-hard for structures beyond trees
 - ➔ Unlikely to find efficient algorithms
 - ➔ We can resort to simple heuristic search
 - Local steps: edge addition, deletion and reversal
 - Augmented hill climbing algorithms to avoid local maxima
 - ➔ There are better algorithms
 - Make larger progress on the search space
 - Computationally more expensive and harder to implement

SCORING FUNCTIONS

Structure Learning Approaches

- Score-based:
 - Define a **scoring function** that evaluates how well a structure matches the data
 - **Search** for a structure that **maximises** the score
 - It amounts to an optimisation problem over the space of network structures

Scoring Functions

How do we decide that a structure is good enough?

1. We can **test** whether the **conditional independence assertions** implicit in the structure are actually **satisfied in the data**.
 - ▶ Because **statistical fluctuations** in the data this test will never be satisfied exactly. (recall the coin tossing example)
 - ▶ We need to perform a suitable statistical test to see if there is sufficient evidence that the independence hypothesis is violated.
 - ▶ The complexity of the resulting network will depend on the threshold used for this test—the stricter the independence test, the more links will be added and the greater the danger of overfitting.
2. **Assess the degree** to which the proposed model **explains the data**

Assessing how well a model explains the data

Likelihood Score

- Recall: the likelihood function measures the probability of the data **given a model**
- It seems intuitive to **find a model** that would make the **data as probable as possible**.
 - ▶ Find a graph **G and parameters θ** that maximise the **likelihood**:
We should find a graph **G** that achieves the **highest likelihood** when we use **MLE parameters for G**
- For a learned graph G , consider **the best possible parameters** (for G) and use that as way to evaluate its quality.

$$\text{score}_L(G | D) = \text{Log}((\hat{\theta}, G) | D)$$

Where $\hat{\theta}$ is the MLE of the parameters given the graph G and the data D .

Example

- Assuming we have **m** data samples: $\mathbf{d}_1, \dots, \mathbf{d}_m$

G_1 :



G_2 :



$$\text{score}_L(G_1 : D) = \sum_{i=1}^m \left(\log \hat{\theta}_{x[\mathbf{d}_i]} + \log \hat{\theta}_{y[\mathbf{d}_i]} \right)$$

this is just the decomposition of the **log likelihood** that we use when we talked about **parameter estimation**.

$$\text{score}_L(G_2 : D) = \sum_{i=1}^m \left(\log \hat{\theta}_{x[\mathbf{d}_i]} + \log \hat{\theta}_{y[\mathbf{d}_i] | x[\mathbf{d}_i]} \right)$$

where $\hat{\theta}_x$ is the MLE for $P(x)$, and $\hat{\theta}_{y|x}$ is the MLE for $P(y | x)$

Decomposition

- The likelihood score decomposes as:

$$\text{score}_L(G : D) = m \sum_{i=1}^n \text{MI}_P(X_i, \text{pa}(X_i)) - m \sum_{i=1}^n \mathbf{H}_p(X_i)$$

mutual information of X, Y .

$$\text{MI}_P(X, Y) = \sum_{x,y \in \text{dom}(X,Y)} P(x, y) \log \frac{P(x, y)}{P(x)P(y)}$$

entropy of variable X

$$\mathbf{H}_p(X) = - \sum_{x \in \text{dom}(X)} P(x) \log P(x)$$

with P the empirical distribution

Information Theoretic Interpretation

- The likelihood score decomposes as:

$$\text{score}_L(G : D) = m \sum_{i=1}^n \text{MI}_p(X_i, \text{pa}(X_i)) - m \sum_{i=1}^n \mathbf{H}_p(X_i)$$

Recall that $\text{MI}(X, Y)$ measures the strength of the dependence between X and Y in P .

Thus, the score of a graph measures the strength of the dependencies between variables and their parents.

Limitations of the likelihood score

- the maximum likelihood almost always prefers more complex structures.

$$MI_P(X; Y \cup Z) \geq MI_P(X; Y),$$

- ❖ **Intuitively:** if Y gives us a certain amount of information about X , adding Z can only give us more information.
- ❖ The mutual information (MI) between a variable and its parents can only go up if we add another parent,
- ❖ Effectively, the MI will go up except in those few cases where we get a conditional independence assertion holding **exactly** in the empirical distribution
- The maximum likelihood network will exhibit a conditional independence only when that independence happens to hold exactly in the empirical distribution.
 - ➡ the likelihood score overfits the training data
 - ➡ Generalisation fails

Summary

- Likelihood score computes log-likelihood of D relative to G , using MLE parameters for G
 - Parameters optimised for D
- + Nice information-theoretic interpretation in terms of (in)dependencies in G
- Guaranteed to overfill the training data if we do not impose constraints.

Bayesian Information Criteria(BIC) Score

Penalising Complexity

$$\text{score}_{BIC}(G | D) = \mathcal{L}(\hat{\theta}_G | D) - \frac{\log M}{2} \text{Dim}[G]$$

$$\text{score}_L(G | D) = \mathcal{L}(\hat{\theta}_G | D)$$

M : the number of training instances

$\text{Dim}[G]$: the number of independent parameters

Tradeoff between fit data and model complexity

Asymptotic Behaviour

$$\begin{aligned}\mathcal{L}(\hat{\theta}_G | D) - \frac{\log M}{2} \text{Dim}[G] &= \\ &= M \sum_{i=1}^n \text{MI}_p(X_i, \text{pa}(X_i)) - M \sum_{i=1}^n \mathbf{H}_p(X_i) - \frac{\log M}{2} \text{Dim}[G]\end{aligned}$$

Asymptotic Behaviour

$$\mathcal{L}(\hat{\theta}_G | D) - \frac{\log M}{2} \text{Dim}[G] =$$
$$= M \sum_{i=1}^n \text{MI}_p(X_i, pa(X_i)) - M \sum_{i=1}^n \mathbf{H}_p(X_i) - \frac{\log M}{2} \text{Dim}[G]$$

Mutual information grows linearly with M while complexity grows logarithmically with M

➡ As M grows more emphasis is given to fit data

Consistency

What network we would learn as the number of samples grows?

$$M \sum_{i=1}^n \text{MI}_p(X_i, \text{pa}(X_i)) - M \sum_{i=1}^n \mathbf{H}_p(X_i) - \frac{\log M}{2} \text{Dim}[G]$$

- Assuming that the data is generated from a particular true structure G^* .
- As M grows the true graph G^* maximises the score

Consistency

What network we would learn as the number of samples grows?

$$M \sum_{i=1}^n \text{MI}_p(X_i, \text{pa}(X_i)) - M \sum_{i=1}^n \mathbf{H}_p(X_i) - \frac{\log M}{2} \text{Dim}[G]$$

- Assuming that the data is generated from a particular true structure G^* .
- As M grows the true graph G^* (**or any other Markov equivalent structure**) maximises the score

Summary


- BIC score explicitly penalises model complexity
- BIC is asymptotically consistent:
 - ❖ If data is generated by a true graph G^* , Markov equivalent networks to G^* will have the highest BIC score as M (the number of data instances) grows to infinity

Bayesian Score

- the Bayesian approach : whenever we have uncertainty over anything, we should place a distribution over it.
- Uncertainty over the structure and the parameters
- Define priors
 - $P(G)$: a prior probability on different graph structures
 - $P(\theta_G | G)$: parameter prior probability on different choice of parameters once the graph G is given
- Using Baye's Rule

$$P(G | D) = \frac{P(D | G)P(G)}{P(D)}$$

Bayesian Score

$$P(G | D) = \frac{P(D | G)P(G)}{P(D)}$$


$P(D)$ does not help to distinguish between different structures. Thus the **Bayesian score** is defined as:

$$\text{score}_B(G | D) = \log P(D | G) + \log P(G)$$

- ❖ a **prior over structures** $P(G)$ gives us a way of **preferring** some structures over others: e.,g penalise dense structures over sparse ones.
- ❖ However, the structure-prior in the score is almost irrelevant compared to $P(D | G)$.
- ❖ $P(D | G)$ takes into consideration our **uncertainty over the parameters**. This is called the **marginal likelihood** of the data given the structure

Marginal Likelihood

marginal likelihood :
$$P(D | G) = \int_{\theta_G} P(D | \theta_G, G) P(\theta_G | G) d\theta_G$$

where $P(D | \theta_G, G)$ is the **likelihood of the data given the network** $\langle G, \theta \rangle$

- the **marginal likelihood** is different from the **maximum likelihood score**
- The **maximum likelihood score** returns the maximum of the likelihood (function) of the data given a structure
- The marginal likelihood is the **average value** of the likelihood (function) of the data given a structure
 - ♣ We average based on the prior measure $P(\theta_G | G)$

Marginal Likelihood

$$P(D | G) = \int_{\theta_G} P(D | \theta_G) P(\theta_G | G) d\theta_G$$

By integrating $P(D | \theta_G)$ over θ_G , we are measuring the average likelihood of the data over different possible choices of θ_G .

- We are being more conservative in our evaluation of the “goodness” of the model.

Recall: the maximal likelihood is overly “optimistic” in its evaluation of the score: It evaluates the likelihood of the training data using the best parameter values for the given data.

➡ thus, avoiding overfitting

Marginal Likelihood

$$P(D | G) = \int_{\theta_G} P(D | \theta_G, G) P(\theta_G | G) d\theta_G$$

The posterior over parameters provides us with a **range of choices**, along with a **measure of how likely each of them is**.

By integrating $P(D | \theta_G, G)$ over the **different choices of parameters** θ_G , we are **measuring the expected likelihood**, averaged over **different possible choices of θ_G** .

- We are being more conservative in our estimate of the “goodness” of the model.

➡ thus, avoiding overfitting

Marginal Likelihood

$$P(D | G) = \int_{\theta_G} P(D | \theta_G) P(\theta_G | G) d\theta_G$$

The posterior over parameters provides us with a **range of choices**, along with a **measure of**

By integrating $P(D | \theta_G)$, we are **measuring the** possible choices of θ_G

The Bayesian approach tells us that, although the choice of MLE parameters is the most likely given the training set D, it is not the only choice.

- We are being more careful of the model.

➡ thus, avoiding overfitting

Marginal likelihood for BN

If we also assume **global parameter independence**,

$$\text{that is, } P(\theta_{X_1}, \dots, \theta_{X_k} \mid G) = \prod_{i=1}^k P(\theta_{X_i} \mid G)$$

as well as, **local parameter independence** that is we can further decompose over the families in the net and values of variables.

then we can simplify the integral in the marginal likelihood using the Gamma function $\Gamma(n)$.

Marginal likelihood for BN

We can rewrite the probability of D given G as a product:

$$P(D | G) = \prod_i^k \prod_{\mathbf{u}_i \in \text{dom}(\text{pa}(X_i))} \frac{\Gamma(\alpha_{X_i})}{\Gamma(\alpha_{X_i|\mathbf{u}_i}) + M[\mathbf{u}_i]} \prod_{x_i^j \in \text{dom}(X_i)} \left[\frac{\Gamma(\alpha_{x_i^j|\mathbf{u}_i} + M[x_i^j, \mathbf{u}_i])}{\Gamma(\alpha_{x_i^j|\mathbf{u}_i})} \right]$$

A product over variables

All possible assignments to the parents of X_i

All possible values x_i^j of variable X_i

Marginal likelihood for BN

We can rewrite the probability of D given G as a product:

$$P(D | G) = \prod_i^k \prod_{\mathbf{u}_i \in \text{dom}(\text{pa}(X_i))} \frac{\Gamma(\alpha_{X_i})}{\Gamma(\alpha_{X_i|\mathbf{u}_i}) + M[\mathbf{u}_i]} \prod_{x_i^j \in \text{dom}(X_i)} \left[\frac{\Gamma(\alpha_{x_i^j|\mathbf{u}_i} + M[x_i^j, \mathbf{u}_i])}{\Gamma(\alpha_{x_i^j|\mathbf{u}_i})} \right]$$

Gamma terms that involve **Dirichlet prior parameters** α ,

Counts over the data $M[\mathbf{u}_i], M[x_i^j, \mathbf{u}_i]$

Summary

- Score-based methods consider the whole structure at once.
- They are therefore less sensitive to individual failures and better at making compromises between independency of variables in the data and the “cost” of adding the edge.
- However, they pose a search problem that may not have an elegant and efficient solution.