

# Winter School: Mathematics for Data Modelling

## Lecture 3: Learning parameters and structure

**Zoubin Ghahramani**

`zoubin@eng.cam.ac.uk`

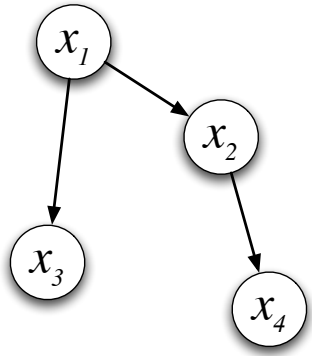
`http://learning.eng.cam.ac.uk/zoubin/`

**Department of Engineering  
University of Cambridge, UK**

**Machine Learning Department  
Carnegie Mellon University, USA**

**Sheffield, January 2008**

# Learning parameters



$$P(x_1)P(x_2|x_1)P(x_3|x_1)P(x_4|x_2)$$

$\theta_2$	$x_2$			
		0.2	0.3	0.5
$x_1$		0.1	0.6	0.3

Assume each variable  $x_i$  is discrete and can take on  $K_i$  values.

The parameters of this model can be represented as 4 tables:  $\theta_1$  has  $K_1$  entries,  $\theta_2$  has  $K_1 \times K_2$  entries, etc.

These are called **conditional probability tables** (CPTs) with the following semantics:

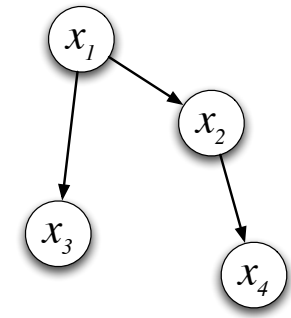
$$P(x_1 = k) = \theta_{1,k} \quad P(x_2 = k'|x_1 = k) = \theta_{2,k,k'}$$

If node  $i$  has  $M$  parents,  $\theta_i$  can be represented either as an  $M + 1$  dimensional table, or as a 2-dimensional table with  $\left(\prod_{j \in \text{pa}(i)} K_j\right) \times K_i$  entries by collapsing all the states of the parents of node  $i$ . Note that  $\sum_{k'} \theta_{i,k,k'} = 1$ .

Assume a data set  $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^N$ .

**How do we learn  $\theta$  from  $\mathcal{D}$ ?**

# Learning parameters



Assume a data set  $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^N$ . How do we learn  $\theta$  from  $\mathcal{D}$ ?

$$P(\mathbf{x}|\theta) = P(x_1|\theta_1)P(x_2|x_1, \theta_2)P(x_3|x_1, \theta_3)P(x_4|x_2, \theta_4)$$

Likelihood:

$$P(\mathcal{D}|\theta) = \prod_{n=1}^N P(\mathbf{x}^{(n)}|\theta)$$

Log Likelihood:

$$\log P(\mathcal{D}|\theta) = \sum_{n=1}^N \sum_i \log P(x_i^{(n)}|x_{\text{pa}(i)}^{(n)}, \theta_i)$$

This decomposes into sum of functions of  $\theta_i$ . Each  $\theta_i$  can be optimized separately:

$$\hat{\theta}_{i,k,k'} = \frac{n_{i,k,k'}}{\sum_{k''} n_{i,k,k''}}$$

where  $n_{i,k,k'}$  is the number of times in  $\mathcal{D}$  where  $x_i = k'$  and  $x_{\text{pa}(i)} = k$ .

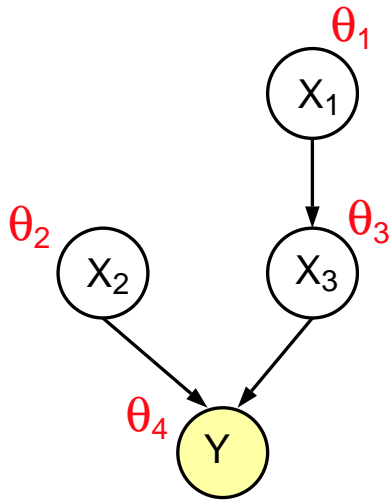
$n_2$	$x_2$	
	2	3
	3	0
$x_1$	3	6

 $\Rightarrow$ 

$\theta_2$	$x_2$	
	0.4	0.6
	0.3	0.6
$x_1$	0.3	0.6

ML solution: **Simply calculate frequencies!**

# Maximum Likelihood Learning with Hidden Variables: The EM Algorithm



Assume a model parameterised by  $\theta$  with observable variables  $Y$  and hidden variables  $X$

**Goal:** maximize parameter log likelihood given observed data.

$$\mathcal{L}(\theta) = \log p(Y|\theta) = \log \sum_X p(Y, X|\theta)$$

# Maximum Likelihood Learning with Hidden Variables: The EM Algorithm

**Goal:** maximise parameter log likelihood given observables.

$$\mathcal{L}(\theta) = \log p(Y|\theta) = \log \sum_X p(Y, X|\theta)$$

The EM algorithm (intuition):

Iterate between applying the following two steps:

- **The E step:** fill-in the hidden/missing variables
- **The M step:** apply complete data learning to filled-in data.

# Maximum Likelihood Learning with Hidden Variables: The EM Algorithm

**Goal:** maximise parameter log likelihood given observables.

$$\mathcal{L}(\theta) = \log p(Y|\theta) = \log \sum_X p(Y, X|\theta)$$

The EM algorithm (derivation):

$$\mathcal{L}(\theta) = \log \sum_X q(X) \frac{p(Y, X|\theta)}{q(X)} \geq \sum_X q(X) \log \frac{p(Y, X|\theta)}{q(X)} = \mathcal{F}(q(X), \theta)$$

- **The E step:** maximize  $\mathcal{F}(q(X), \theta^{[t]})$  wrt  $q(X)$  holding  $\theta^{[t]}$  fixed:

$$q(X) = P(X|Y, \theta^{[t]})$$

- **The M step:** maximize  $\mathcal{F}(q(X), \theta)$  wrt  $\theta$  holding  $q(X)$  fixed:

$$\theta^{[t+1]} \leftarrow \operatorname{argmax}_{\theta} \sum_X q(X) \log p(Y, X|\theta)$$

The E-step requires solving the *inference* problem, finding the distribution over the hidden variables  $p(X|Y, \theta^{[t]})$  given the current model parameters. This can be done using **belief propagation** or the **junction tree algorithm**.

# Maximum Likelihood Learning with Hidden Variables: The EM Algorithm

## ML Learning with Complete Data (No Hidden Variables)

Log likelihood decomposes into sum of functions of  $\theta_i$ . Each  $\theta_i$  can be optimized separately:

$$\hat{\theta}_{ijk} \leftarrow \frac{n_{ijk}}{\sum_{k'} n_{ijk'}}$$

where  $n_{ijk}$  is the number of times in  $\mathcal{D}$  where  $x_i = k$  and  $x_{\text{pa}(i)} = j$ .

Maximum likelihood solution: **Simply calculate frequencies!**

## ML Learning with Incomplete Data (i.e. with Hidden Variables)

Iterative EM algorithm

**E step:** compute expected counts given previous settings of parameters  $E[n_{ijk} | \mathcal{D}, \boldsymbol{\theta}^{[t]}]$ .

**M step:** re-estimate parameters using these expected counts

$$\theta_{ijk}^{[t+1]} \leftarrow \frac{E[n_{ijk} | \mathcal{D}, \boldsymbol{\theta}^{[t]}]}{\sum_{k'} E[n_{ijk'} | \mathcal{D}, \boldsymbol{\theta}^{[t]}]}$$

# Bayesian parameter learning with no hidden variables

Let  $n_{ijk}$  be the number of times  $(x_i^{(n)} = k \text{ and } x_{\text{pa}(i)}^{(n)} = j)$  in  $\mathcal{D}$ .  
For each  $i$  and  $j$ ,  $\theta_{ij\cdot}$  is a probability vector of length  $K_i \times 1$ .

Since  $x_i$  is a discrete variable with probabilities given by  $\theta_{i,j,\cdot}$ , the likelihood is:

$$P(\mathcal{D}|\boldsymbol{\theta}) = \prod_n \prod_i P(x_i^{(n)} | x_{\text{pa}(i)}^{(n)}, \boldsymbol{\theta}) = \prod_i \prod_j \prod_k \theta_{ijk}^{n_{ijk}}$$

If we choose a prior on  $\boldsymbol{\theta}$  of the form:

$$P(\boldsymbol{\theta}) = c \prod_i \prod_j \prod_k \theta_{ijk}^{\alpha_{ijk}-1}$$

where  $c$  is a normalization constant, and  $\sum_k \theta_{ijk} = 1 \forall i, j$ , then the posterior distribution also has the same form:

$$P(\boldsymbol{\theta}|\mathcal{D}) = c' \prod_i \prod_j \prod_k \theta_{ijk}^{\tilde{\alpha}_{ijk}-1}$$

where  $\tilde{\alpha}_{ijk} = \alpha_{ijk} + n_{ijk}$ .

This distribution is called the **Dirichlet distribution**.



# Dirichlet Distribution

The **Dirichlet distribution** is a distribution over the  $K$ -dim probability simplex.

Let  $\boldsymbol{\theta}$  be a  $K$ -dimensional vector s.t.  $\forall j : \theta_j \geq 0$  and  $\sum_{j=1}^K \theta_j = 1$

$$P(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \text{Dir}(\alpha_1, \dots, \alpha_K) \stackrel{\text{def}}{=} \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)} \prod_{j=1}^K \theta_j^{\alpha_j-1}$$

where the **first term** is a normalization constant<sup>1</sup> and  $E(\theta_j) = \alpha_j / (\sum_k \alpha_k)$

The Dirichlet is **conjugate to the multinomial distribution**. Let

$$x|\boldsymbol{\theta} \sim \text{Multinomial}(\cdot|\boldsymbol{\theta})$$

That is,  $P(x = j|\boldsymbol{\theta}) = \theta_j$ . Then the posterior is also Dirichlet:

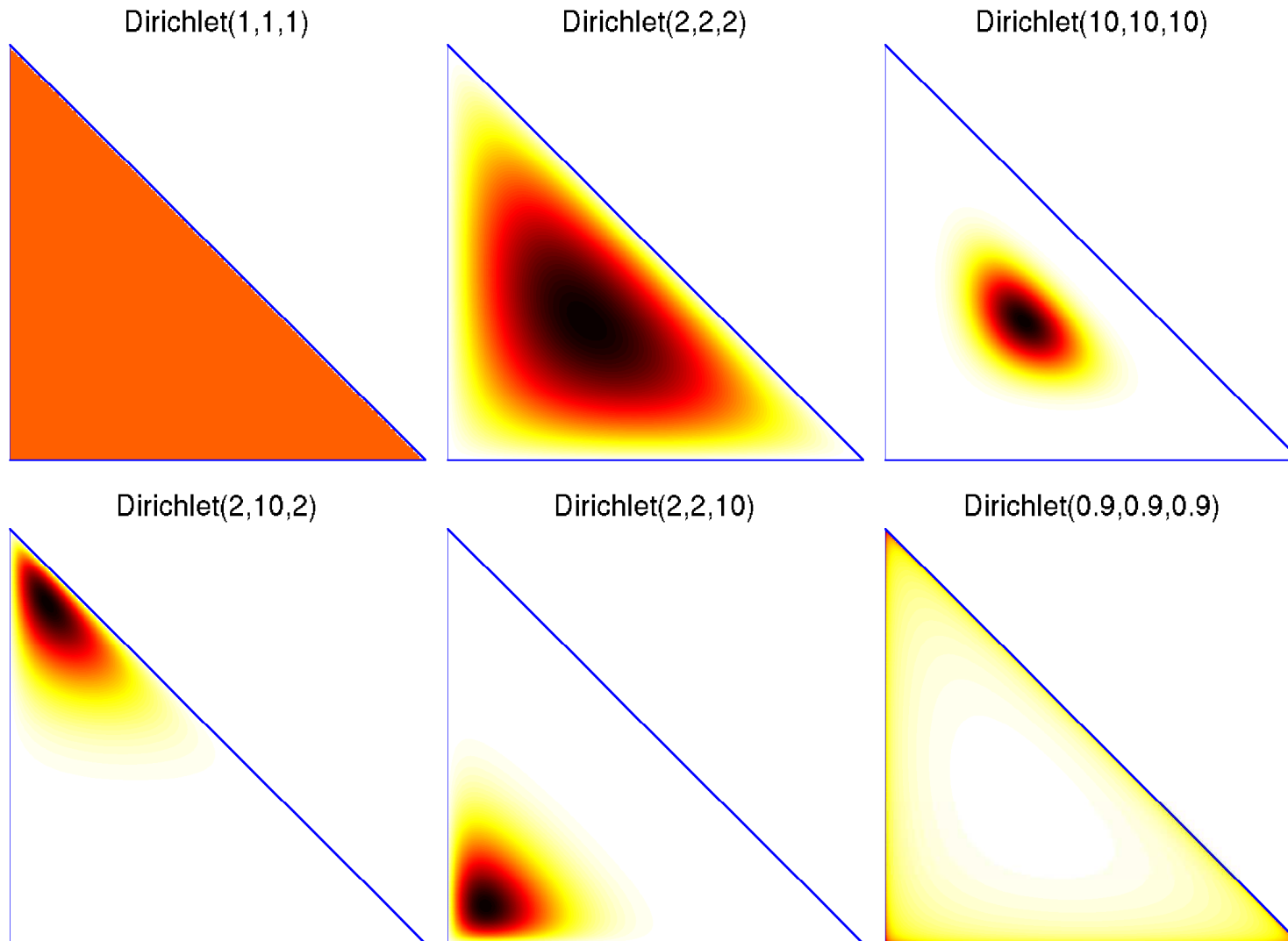
$$P(\boldsymbol{\theta}|x = j, \boldsymbol{\alpha}) = \frac{P(x = j|\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\alpha})}{P(x = j|\boldsymbol{\alpha})} = \text{Dir}(\tilde{\boldsymbol{\alpha}})$$

where  $\tilde{\alpha}_j = \alpha_j + 1$ , and  $\forall \ell \neq j : \tilde{\alpha}_\ell = \alpha_\ell$

<sup>1</sup> $\Gamma(x) = (x-1)\Gamma(x-1) = \int_0^\infty t^{x-1}e^{-t}dt$ . For integer  $n$ ,  $\Gamma(n) = (n-1)!$

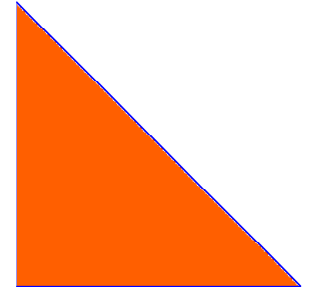
# Dirichlet Distributions

Examples of Dirichlet distributions over  $\theta = (\theta_1, \theta_2, \theta_3)$  which can be plotted in 2D since  $\theta_3 = 1 - \theta_1 - \theta_2$ :



# Example

Dirichlet(1,1,1)



Assume  $\alpha_{ijk} = 1 \forall i, j, k$ .

This corresponds to a **uniform** prior distribution over parameters  $\theta$ . This is not a very strong/dogmatic prior, since any parameter setting is assumed a priori possible.

After observed data  $\mathcal{D}$ , what are the parameter posterior distributions?

$$P(\theta_{ij.} | \mathcal{D}) = \text{Dir}(n_{ij.} + 1)$$

This distribution predicts, for future data:

$$P(x_i = k | x_{\text{pa}(i)} = j, \mathcal{D}) = \frac{n_{ijk} + 1}{\sum_{k'} (n_{ijk'} + 1)}$$

Adding 1 to each of the counts is a form of smoothing called **“Laplace’s Rule”**.

# Bayesian parameter learning with hidden variables

**Notation:** let  $\mathcal{D}$  be the observed data set,  $\mathcal{X}$  be hidden variables, and  $\theta$  be model parameters. Assume discrete variables and Dirichlet priors on  $\theta$

**Goal:** to infer  $P(\theta|\mathcal{D}) = \sum_{\mathcal{X}} P(\mathcal{X}, \theta|\mathcal{D})$

**Problem:** since (a)

$$P(\theta|\mathcal{D}) = \sum_{\mathcal{X}} P(\theta|\mathcal{X}, \mathcal{D})P(\mathcal{X}|\mathcal{D}),$$

and (b) for every way of filling in the missing data,  $P(\theta|\mathcal{X}, \mathcal{D})$  is a Dirichlet distribution, and (c) there are exponentially many ways of filling in  $\mathcal{X}$ , it follows that  $P(\theta|\mathcal{D})$  is a mixture of Dirichlets with exponentially many terms!

**Solutions:**

- Find a single best (“Viterbi”) completion of  $\mathcal{X}$  (Stolcke and Omohundro, 1993)
- Markov chain Monte Carlo methods
- Variational Bayesian methods (Beal and Ghahramani, 2003)

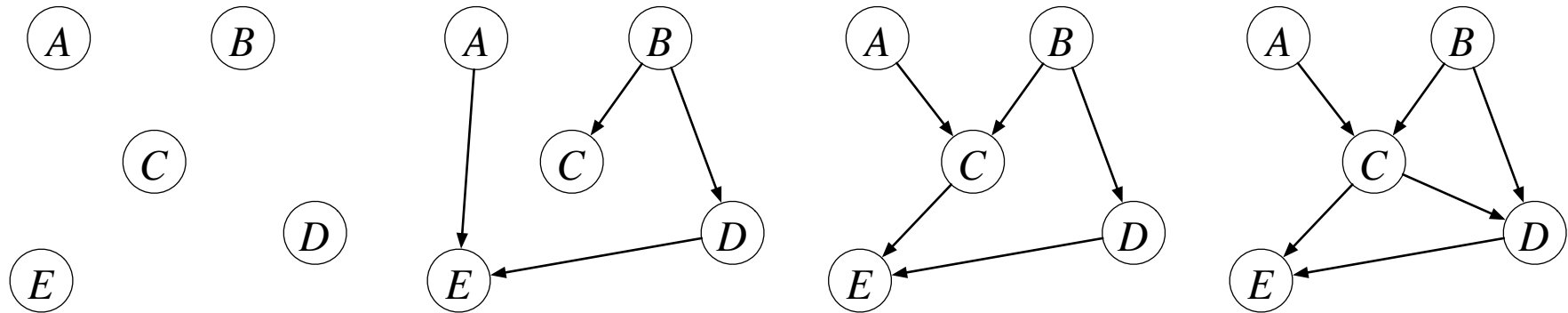
## Summary of parameter learning

	Complete (fully observed) data	Incomplete (hidden / missing) data
ML	calculate frequencies	EM
Bayesian	update Dirichlet distributions	MCMC / Viterbi / VBEM

- For complete data Bayesian learning is not more costly than ML
- For incomplete data VBEM  $\approx$  EM time complexity
- Other parameter priors are possible but Dirichlet is pretty flexible and intuitive.
- For binary data, other parametrizations include:
  - Sigmoid:
 
$$P(x_i = 1 | x_{\text{pa}(i)}, \theta_i) = 1 / (1 + \exp\{-\theta_{i0} - \sum_{j \in \text{pa}(i)} \theta_{ij} x_j\})$$
  - Noisy-or:
 
$$P(x_i = 1 | x_{\text{pa}(i)}, \theta_i) = 1 - \exp\{-\theta_{i0} - \sum_{j \in \text{pa}(i)} \theta_{ij} x_j\}$$
- For non-discrete data, similar ideas but generally harder inference and learning.

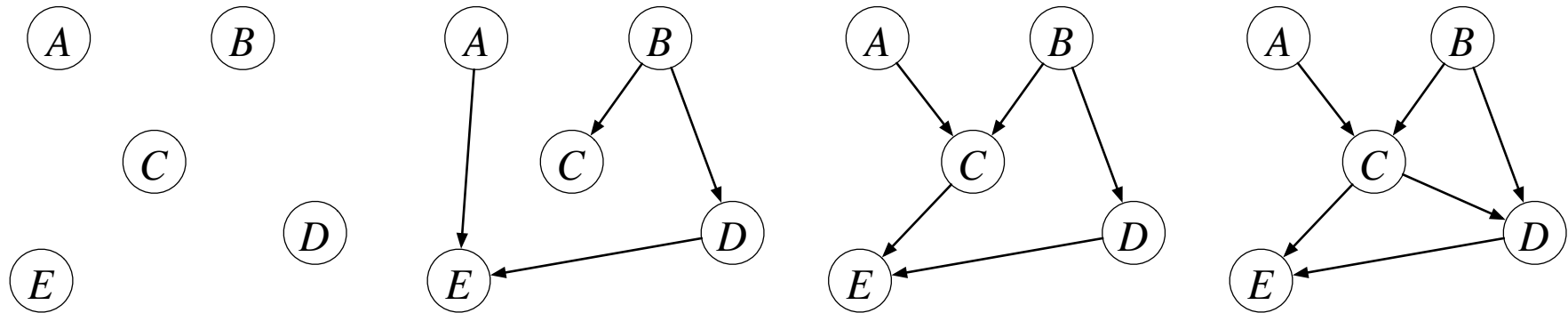
# Structure learning

Given a data set of observations of  $(A, B, C, D, E)$  can we learn the structure of the graphical model?



Let  $m$  denote the graph structure = the set of edges.

# Structure learning



**Constraint-Based Learning:** Use statistical tests of marginal and conditional independence. Find the set of DAGs whose d-separation relations match the results of conditional independence tests.

**Score-Based Learning:** Use a global score such as the BIC score or Bayesian marginal likelihood. Find the structures that maximize this score.

# Score-based structure learning for complete data

Consider a graphical model with structure  $m$ , discrete observed data  $\mathcal{D}$ , and parameters  $\theta$ . Assume Dirichlet priors.

The Bayesian marginal likelihood score is easy to compute:

$$\text{score}(m) = \log P(\mathcal{D}|m) = \log \int P(\mathcal{D}|\theta, m)P(\theta|m)d\theta$$

$$\text{score}(m) = \sum_i \sum_j \left[ \log \Gamma\left(\sum_k \alpha_{ijk}\right) - \sum_k \log \Gamma(\alpha_{ijk}) - \log \Gamma\left(\sum_k \tilde{\alpha}_{ijk}\right) + \sum_k \log \Gamma(\tilde{\alpha}_{ijk}) \right]$$

where  $\tilde{\alpha}_{ijk} = \alpha_{ijk} + n_{ijk}$ . **Note that the score decomposes over  $i$ .**

One can incorporate structure prior information  $P(m)$  as well:

$$\text{score}(m) = \log P(\mathcal{D}|m) + \log P(m)$$

**Greedy search algorithm:** Start with  $m$ . Consider modifications  $m \rightarrow m'$  (edge deletions, additions, reversals). Accept  $m'$  if  $\text{score}(m') > \text{score}(m)$ . Repeat.

**Bayesian inference of model structure:** Run MCMC on  $m$ .



# Bayesian Structural EM for *incomplete* data

Consider a graphical model with structure  $m$ , observed data  $\mathcal{D}$ , hidden variables  $\mathcal{X}$  and parameters  $\theta$

The Bayesian score is generally intractable to compute:

$$\text{score}(m) = P(\mathcal{D}|m) = \int \sum_{\mathcal{X}} P(\mathcal{X}, \theta, \mathcal{D}|m) d\theta$$

**Bayesian Structure EM** (Friedman, 1998):

1. compute MAP parameters  $\hat{\theta}$  for current model  $m$  using EM
2. find hidden variable distribution  $P(\mathcal{X}|\mathcal{D}, \hat{\theta})$
3. for a small set of candidate structures compute or approximate

$$\text{score}(m') = \sum_{\mathcal{X}} P(\mathcal{X}|\mathcal{D}, \hat{\theta}) \log P(\mathcal{D}, \mathcal{X}|m')$$

4.  $m \leftarrow m'$  with highest score

# Directed Graphical Models and Causality

Causal relationships are a fundamental component of cognition and scientific discovery.

Even though the independence relations are identical, there is a **causal** difference between

- “smoking”  $\rightarrow$  “yellow teeth”
- “yellow teeth”  $\rightarrow$  “smoking”

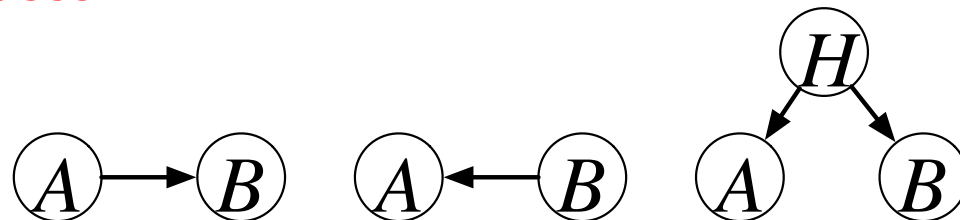
**Key idea:** interventions and the do-calculus:

$$P(S|Y = y) \neq P(S|\text{do}(Y = y))$$

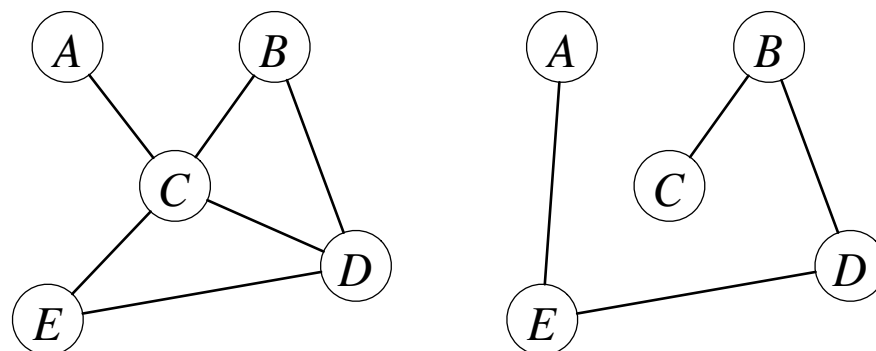
$$P(Y|S = s) = P(Y|\text{do}(S = s))$$

Causal relationships are robust to interventions on the parents.

The **key difficulty** in learning causal relationships from observational data is the presence of **hidden common causes**:



# Learning parameters and structure in undirected graphs



$$P(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_j g_j(\mathbf{x}_{C_j}; \boldsymbol{\theta}_j) \text{ where } Z(\boldsymbol{\theta}) = \sum_{\mathbf{x}} \prod_j g_j(\mathbf{x}_{C_j}; \boldsymbol{\theta}_j).$$

**Problem:** computing  $Z(\boldsymbol{\theta})$  is computationally intractable for general (non-tree-structured) undirected models. Therefore, maximum-likelihood learning of parameters is generally intractable, Bayesian scoring of structures is intractable, etc.

## Solutions:

- directly approximate  $Z(\boldsymbol{\theta})$  and/or its derivatives (cf. Boltzmann machine learning; contrastive divergence; pseudo-likelihood)
- use approx inference methods (e.g. loopy belief propagation, bounding methods, EP).

See: (Murray and Ghahramani, 2004; Murray et al, 2006) for Bayesian learning in undirected models.

# Summary

- Parameter learning in directed models:
  - complete and incomplete data;
  - ML and Bayesian methods
- Structure learning in directed models: complete and incomplete data
- Causality
- Parameter and Structure learning in undirected models