Graphical Model III

Learning

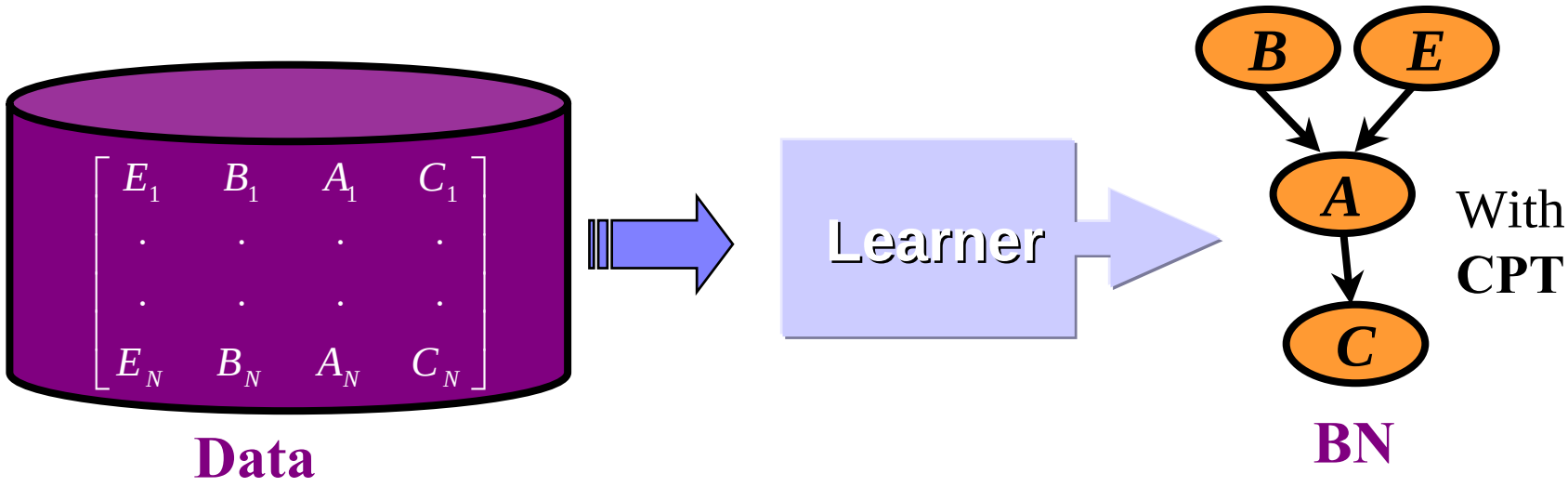
Wang Houfeng
Institute of Computational Linguistics
Peking University

Outline

- **Overview**
- Parameter Learning with full data
- Parameter Learning with partial data
- Structure+Parameter Learning with full data
- Learning parameter of MRF

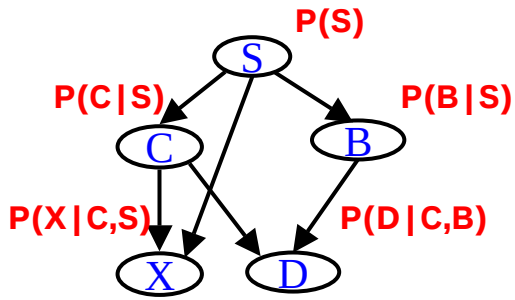
Tasks

- Given training set $D = \{D_1, ..., D_M\}$ with $D_m = (x_1^m, x_2^m, ..., x_n^m)$
  Where x_j stands for the value of Random Variable X_j
- Find GM that best matches ***D***
  - Model (Structure) Learning
  - Parameter(CPT) estimation

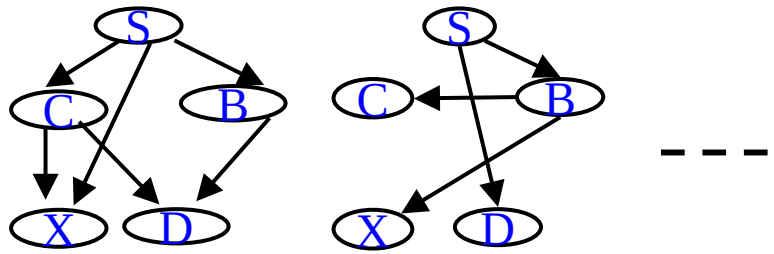


Tasks

- Known Structure – learn parameters(CPT)
  - Full(complete) observable data:
    parameter estimation (ML, MAP)
  - Partial observable data:
    non-linear parametric
    optimization (gradient descent, EM)



- Unknown Structure – learn structure and parameters
  - Full data:
    optimization (search in space of graphs)
  - Partial data(*):
    structural EM,
    mixture models



$$\hat{S} = \arg\max_{S} Score(S)$$

Learning: Two Problems

- Two problems in Bayesian Net:
  1. The graph topology (structure)
  2. The parameters of each CPD (Conditional Prob. Distr)
- Learning:
  - Learning structure (much harder)
  - Learning parameters

Structure	Observability(Data)	Method
Known	Full	Maximum Likelihood Estimation
Known	Partial	EM (or gradient ascent)
Unknown	Full	Search through model space
Unknown	Partial	EM + search through model space

Structure & Parameter

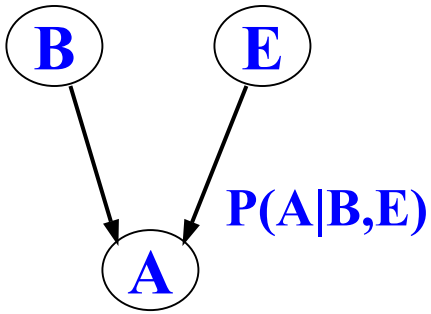
- **The structure of the BN** typically reflects causal relations
  - BNs are also sometime referred to as **causal networks**
  - Causal structure is very intuitive **in many applications** domain and it is relatively easy to obtain from the **domain expert**
- **Parameters of BN** correspond to conditional distributions relating a random variable and its parents
  - Their complexity much smaller than the full joint
  - Easier to come up (estimate) the probabilities from expert or automatically by learning from data

Outline

- Overview
- **➢ Parameter Learning with full data**
- Parameter Learning with partial data
- Structure+Parameter Learning with full data
- Learning parameter of MRF

Known structure+ full observability

- Learning of parameters (CPT)
  - **Idea:** decompose the estimation problem for the full joint over a large number of variables to a set of smaller estimation problems corresponding to parent-variable conditionals.
  - **Example:** Assume A,E,B are binary with *True*, *False* values



6 estimations:

$$P(A \mid B = T, E = T)$$
$$P(A \mid B = T, E = F)$$
$$P(A \mid B = F, E = T)$$
$$P(A \mid B = F, E = F)$$
$$P(B = T)$$
$$P(E = T)$$

Full Observed Data

- The Observed Training Set D:

$$D = \{D_1, ..., D_M\} \text{ with } D_m = (x_1^m, x_2^m, ..., x_n^m)$$

- For each sample point D_m , the values of all random variable will be observed.

Known structure+ full observability

- Goal: estimate BN parameters θ (**CPT**)
  - All parameters: P(X | Parents(X))
- A parameterization θ is good if it is likely to generate the observed data:

$$L(\theta : D) = P(D \mid \theta) = \prod_m P(D_m \mid \theta)$$

i.i.d. samples

- Maximum Likelihood Estimation (MLE) Principle: Choose θ^* so as to maximize L

Recall: MLE

- Given
  - A sample set $D=\{D_1, \ldots, D_M\}$
  - A vector of parameters (or single para): θ
- Define:
  - Likelihood of the data: $P(D \mid \theta)$
  - **Log-likelihood** of the data: $L(\theta)=\log P(D|\theta)$
- Given D, find

$$\theta_{ML} = \arg\max_{\theta \in \Omega} L(\theta)$$

MLE

- Often we assume that D_is are independently identically distributed (i.i.d.)

$$\begin{aligned}
\theta_{ML} &= \arg\max_{\theta\in\Omega} L(\theta) \\
&= \arg\max_{\theta\in\Omega} \log P(\mathrm{D}\,|\,\theta) \\
&= \arg\max_{\theta\in\Omega} \log P(\mathrm{D}_1,...,D_M\,|\,\theta) \\
&= \arg\max_{\theta\in\Omega} \log \prod_m P(D_m\,|\,\theta) \\
&= \arg\max_{\theta\in\Omega} \sum_m \log P(D_m\,|\,\theta)
\end{aligned}$$

- Depending on the form of p(D|θ), to solve optimization problem.

Parameter estimation

- The MLE (maximum likelihood estimate) solution:
  - for each value *x* of a node *X*
  - and each instantiation ***u*** of *Parents(X)*

$$\theta^{*}_{x|u} = \frac{N(x,u)}{N(u)}$$

sufficient statistics

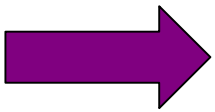
  - Just need to collect the counts for every combination of parents and children observed in the data

An easy example

- Assuming
  - A coin has a probability p of heads, 1-p of tails.
  - Observation: We toss a coin N times, and the result is a set of Hs and Ts, and there are M Hs.
- What is the value of p based on MLE, given the observation?

$$L(\theta) = \log P(D \mid \theta) = \log[\, p^M (1-p)^{N-M} \,]$$
$$= M \log p + (N-M)\log(1-p)$$

$$\frac{dL(\theta)}{dp} = \frac{d(M\log p + (N-M)\log(1-p))}{dp} = \frac{M}{p} - \frac{N-M}{1-p} = 0$$

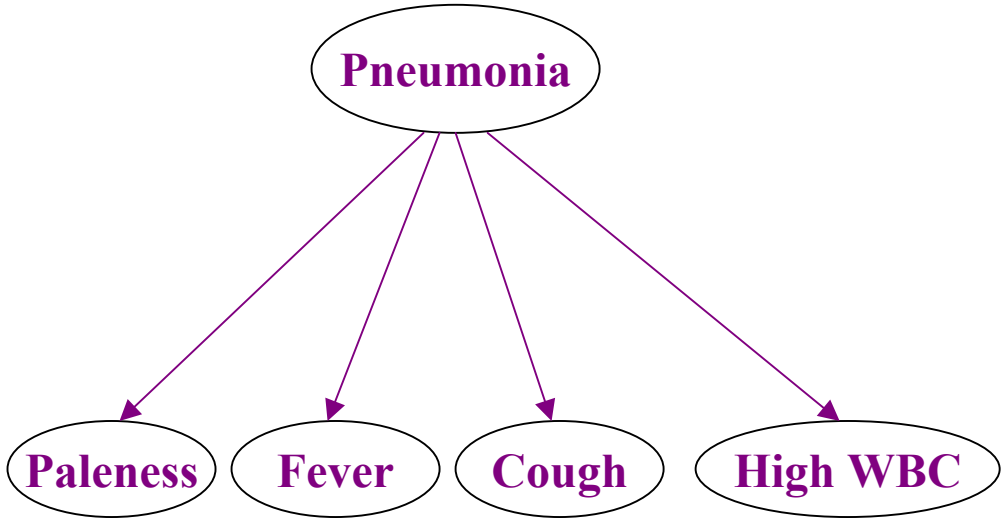


p= M/N

Example

Data D (different patient cases):

Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F



Example:

Estimating: P(Fever|Pneumonia=T)

Learning: P(Fever|Pneumonia=T)

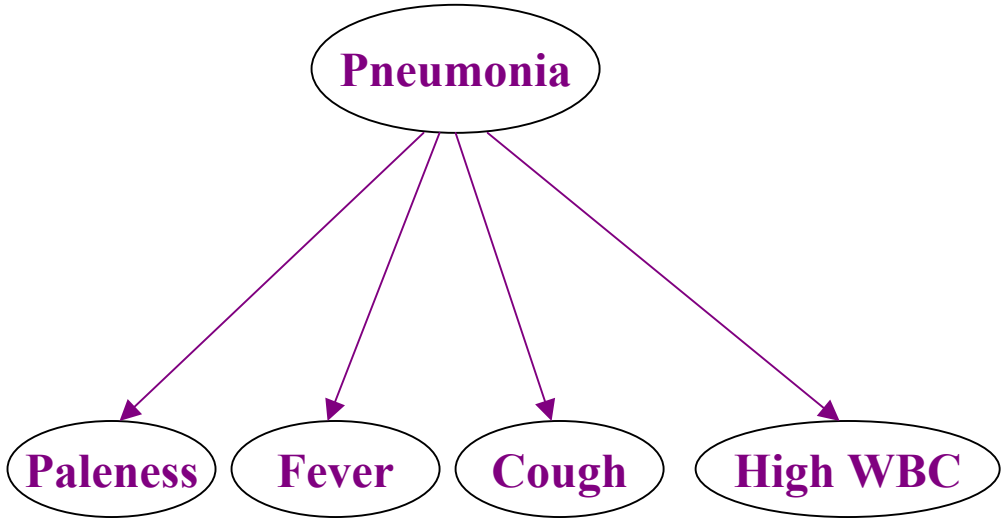
- Select data points with Pneumonia=T

  (ignore the rest)
- Focus on (select) only values of the random variable defining the distribution (Fever)
- Learn the parameters of the conditional the same way as we learned the parameters of the biased coin or dice

Step 1: Selecting Pneumonia

Data D (different patient cases):

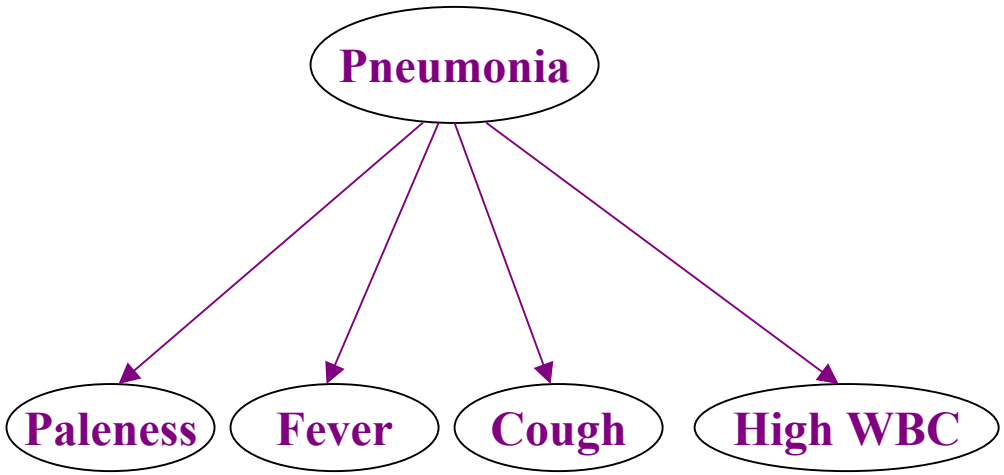
Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F



Step 2: Selecting Fever

Data D (different patient cases):

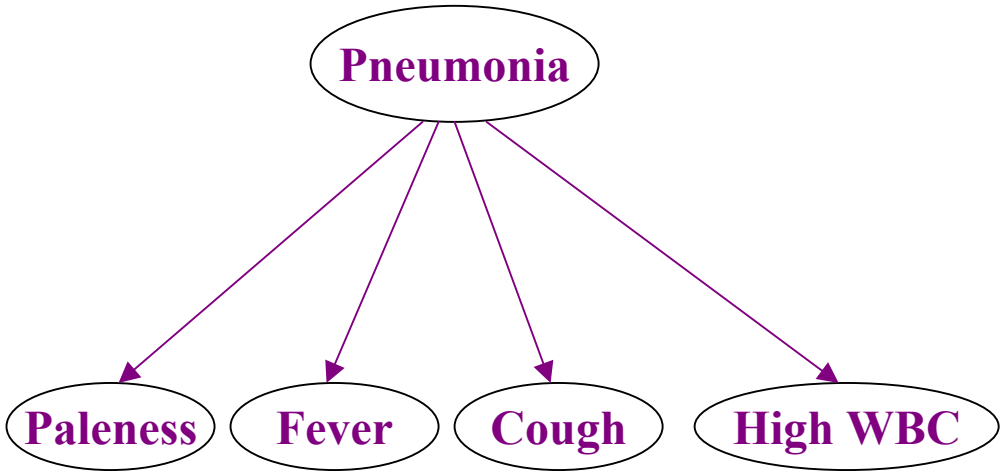
Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F



Step: Learning Parameter

- **Learning the ML estimate:**

Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F



P(Fever|Pneumonia=T)

T	F
0.6	0.4

Outline

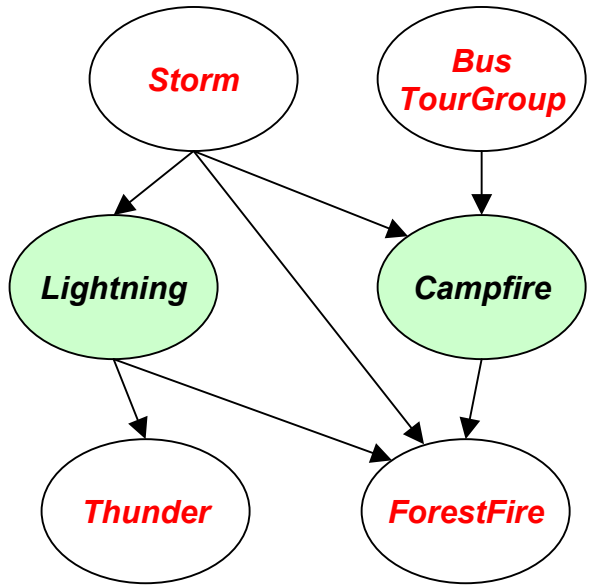
- Overview
- Parameter Learning with full data
- **Parameter Learning with partial data**
- Structure+Parameter Learning with full data
- Learning parameter of MRF

Known structure+**Partial** Observability

- **Variables which are always unobserved are called hidden variables.**
- If variables are occasionally unobserved they are missing data.

– Example
  - Can observe *ForestFire*, *Storm*, *BusTourGroup*, *Thunder*
  - Can't observe *Lightning*, *Campfire*

– Similar to training artificial neural net with hidden units
  - Causes: *Storm*, *BusTourGroup*
  - Observable effects: *ForestFire*, *Thunder*
  - Intermediate variables: *Lightning*, *Campfire*



Partial Observability

- This missing value (hidden variable) problem arises frequently.
- Chicken-and-Egg issue: if we had CPTs, we could fill in the data, or if we had data we could estimate CPTs.
- We *do* have partial data and partial (prior) CPTs. Can we somehow leverage these into full data and posterior CPTs?

General View

- Maximize posterior probability (or likelihood if uniform priors).
- Let $\theta = (w_1,\ldots,w_k)$ are the probabilities in the CPTs (analogous to weights in a neural network)
- Let $D=\{ D_1,\ldots, D_M \}$ is the data set
- Use a greedy hill-climbing search (making small changes in the direction of the gradient) to maximize

$$\begin{aligned} P_\theta(D) &= P(D|\theta) \\ &= P(D|w_1,\ldots,w_k) \\ &= P(D_1|w_1,\ldots,w_k)\ldots P(D_M|w_1,\ldots,w_k) \\ &= \Pi\, P(D_i|w_1,\ldots,w_k) \end{aligned}$$

- it is easier to do this with the logarithm of likelihood

Gradient Ascent

- Let w_{ijk} denote an entry in one of CPT: the random var is Y_i and its value is y_{ij}, its Parents are U_i and the value is u_{ik}.

For example, Y_i=Campfire, U_i=<*Storm*, *BusTourGroup*>

The values: y_{ij}=*True*, and u_{ik} = < *False*, *False* >

$$\frac{\partial \ln P_\theta(D)}{\partial w_{ijk}} = \frac{\partial \ln \prod_{D_m \in D} P_\theta(D_m)}{\partial w_{ijk}}$$

$$= \sum_{D_m \in D} \frac{\partial \ln P_\theta(D_m)}{\partial w_{ijk}}$$

$$= \sum_{D_m \in D} \frac{1}{P_\theta(D_m)} \frac{\partial P_\theta(D_m)}{\partial w_{ijk}}$$

(Note: $\frac{\partial \ln f(x)}{\partial x} = \frac{1}{f(x)} \frac{\partial f(x)}{\partial x}$)

Gradient Ascent

- Find the gradient contribution for a single case (D_j) from a single CPT with which w_{ijk} is associated.
- Now, introduce the values of the variable Y_i and its parents $U_i = Parents(Y_i)$ by summing over their possible value $y_{ij'}$ and $u_{ik'}$

$$\frac{\partial \ln P_\theta(D)}{\partial w_{ijk}} = \sum_{D_m \in D} \frac{1}{P_\theta(D_m)} \frac{\partial P_\theta(D_m)}{\partial w_{ijk}}$$

$$= \sum_{D_m \in D} \frac{1}{P_\theta(D_m)} \frac{\partial}{\partial w_{ijk}} \sum_{j',k'} P_\theta(D_m \mid y_{ij'}, u_{ik'}) P_\theta(y_{ij'}, u_{ik'})$$

$$= \sum_{D_m \in D} \frac{1}{P_\theta(D_m)} \frac{\partial}{\partial w_{ijk}} \sum_{j',k'} P_\theta(D_m \mid y_{ij'}, u_{ik'}) P_\theta(y_{ij'} \mid u_{ik'}) P_\theta(u_{ik'})$$

Gradient Ascent

- Consider that $w_{ijk} = P_\theta(y_{ij} \mid u_{ik})$, the only term in the sum for which $\frac{\partial}{\partial w_{i\,jk}}$ is nonzero is the term for which $j' = j$ and $i' = i$. *So*,

$$\frac{\partial \ln P_\theta(D)}{\partial w_{ijk}} = \sum_{D_m \in D} \frac{1}{P_\theta(D_m)} \frac{\partial}{\partial w_{ijk}} \sum_{j',k'} P_\theta(D_m \mid y_{ij'}, u_{ik'}) P_\theta(y_{ij'} \mid u_{ik'}) P_\theta(u_{ik'})$$

$$= \sum_{D_m \in D} \frac{1}{P_\theta(D_i)} \frac{\partial}{\partial w_{ijk}} P_\theta(D_m \mid y_{ij}, u_{ik}) w_{ijk} P_\theta(u_{ik})$$

$$= \sum_{D_m \in D} \frac{1}{P_\theta(D_m)} P_\theta(D_i \mid y_{ij}, u_{ik}) P_\theta(u_{ik})$$

$$= \sum_{D_m \in D} \frac{1}{P_\theta(D_m)} \frac{P_\theta(y_{ij}, u_{ik} \mid D_m) P_\theta(D_m) P_\theta(u_{ik})}{P_\theta(y_{ij}, u_{ik})} \quad \text{(Bayes Rule)}$$

$$= \sum_{D_m \in D} \frac{P_\theta(y_{ij}, u_{ik} \mid D_m)}{P_\theta(y_{ij} \mid u_{ik})} = \sum_{D_m \in D} \frac{P_\theta(y_{ij}, u_{ik} \mid D_m)}{w_{ijk}}$$

Learning Bayesian Networks: Gradient Ascent

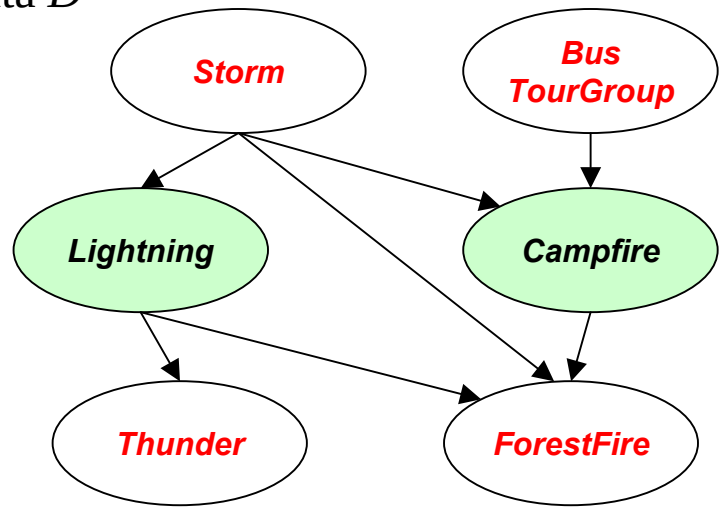
- Algorithm *Train-BN* (*D*)
  - Let w_{ijk} denote one entry in the CPT for variable Y_i in the network
    - $w_{ijk} = P(Y_i = y_{ij} \mid \text{pa}(Y_i) = \text{<the list } u_{ik} \text{ of values>})$
    - e.g., if $Y_i \equiv Campfire$, then (for example) $u_{ik} \equiv$ <*Storm* = *T*, *BusTourGroup* = *F*>
  - WHILE termination condition not met DO // perform gradient ascent
    - Update all CPT entries w_{ijk} using training data *D*

$$w_{ijk} \leftarrow w_{ijk} + r \sum_{D_m \in D} \frac{P_\theta(y_{ij}, u_{ik} \mid D_m)}{w_{ijk}}$$

    - Renormalize w_{ijk} to assure invariants:

$$\sum_j w_{ijk} = 1$$

$$\forall j . 0 \le w_{ijk} \le 1$$



Learning Bayesian Networks: Missing Observations

- Problem Definition

  A set of random variables: $\mathbf{X}=\{X_1,X_2,...,X_n\}$

  Observable Data: $D=\{D_1, D_2, ...,D_M\}$

  But, some values are missing:

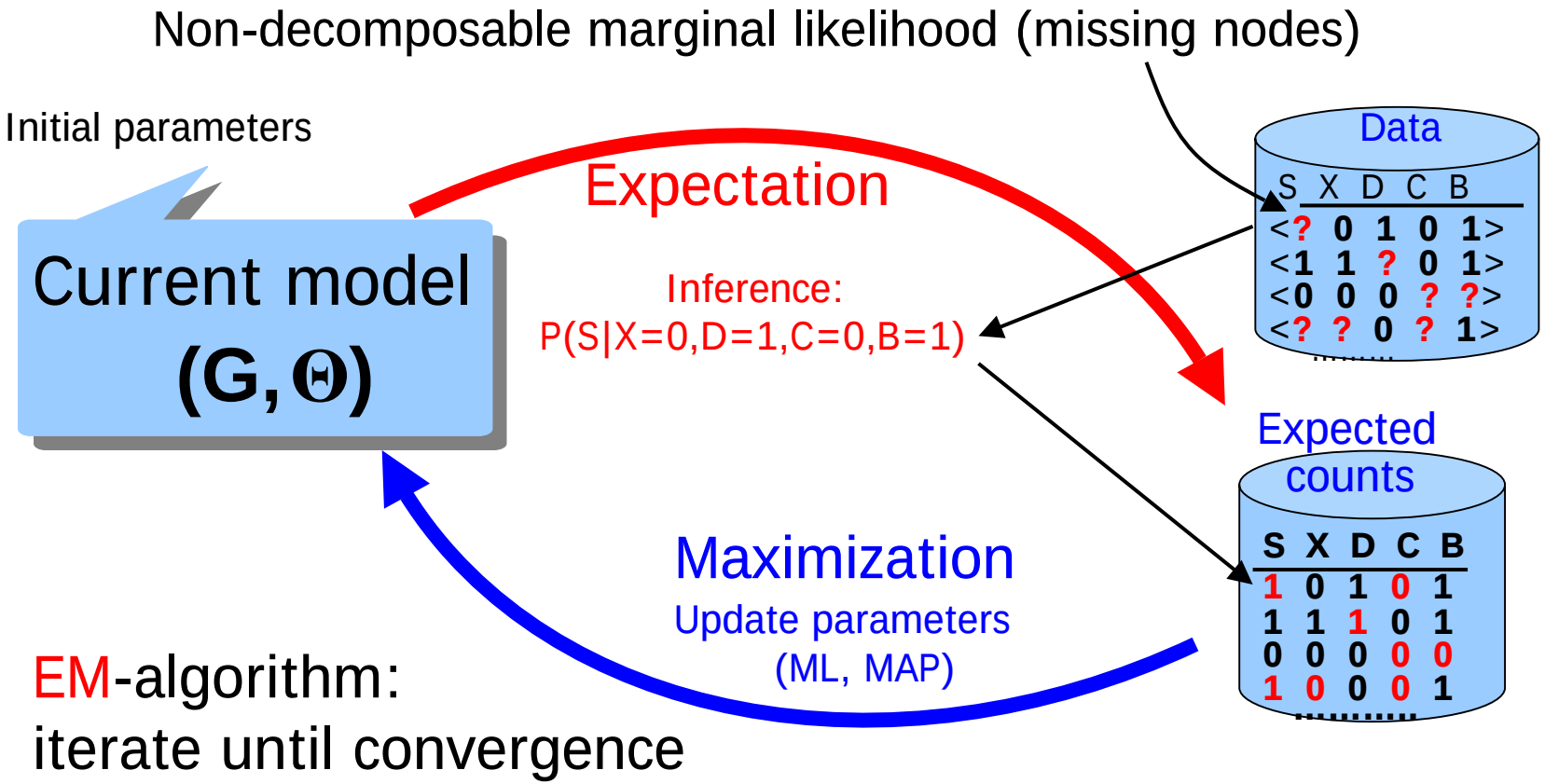
  $D_m=(x_1^m, x_3^m, ..., x_n^m)$ where, x_2^m is missing

  $D_{m+1}=(x_3^{m+1}, ..., x_n^{m+1})$ where, x_1^{m+1} and x_2^{m+1} are missing, ...

  - If variables are occasionally unobserved they are missing data.
  - Variables which are always unobserved are called latent(hidden) variables

- Solution Approaches
  - Expectation-Maximization (EM) algorithm can be used here

Missing Observations



Basic setting in EM

- D is a set of data points: **observed** data
- Θ is a parameter vector.
- EM is a method to find θ_{ML} where

$$\theta_{ML} = \arg\max_{\theta \in \Omega} L(\theta)$$
$$= \arg\max_{\theta \in \Omega} \log P(D \mid \theta)$$

- Calculating P(D | θ) directly is hard.
- Calculating P(D,Y|θ) is much simpler, where Y is "missing" data (or "hidden" data).

The basic EM strategy

- Z = (D, Y)
  - Z: complete data (“augmented data”)
  - D: observed data (“incomplete” data)
  - Y: missing data (“hidden” data)
- Given a fixed D_m, there could be many possible y’s.
  - Ex: given a sentence D_m, there could be many state sequences in an HMM that generates D_m.
- EM algorithm
  - Consider a set of starting parameters
  - Use these parameters to “estimate” the missing data
  - Use “complete” data to update parameters
  - Repeat until convergence

New log-likelihood function

- L is a function of θ, while holding D constant:

$$L(\theta) = P(D \mid \theta)$$

$$l(\theta) = \log L(\theta) = \log P(D \mid \theta)$$

$$= \log \prod_{m=1}^{M} P(D_m \mid \theta)$$

$$= \sum_{m=1}^{M} \log P(D_m \mid \theta)$$

$$= \sum_{m=1}^{M} \log \sum_{y} P(D_m, y \mid \theta)$$

The iterative approach for MLE

$$\theta_{ML} = \arg\max_{\theta \in \Omega} L(\theta)$$
$$= \arg\max_{\theta \in \Omega} l(\theta)$$
$$= \arg\max_{\theta \in \Omega} \sum_{m=1}^{M} \log \sum_{y} p(D_m, y \mid \theta)$$

In many cases, we cannot find the solution directly.

An alternative is to find a sequence: $\theta^0, \theta^1, ..., \theta^t, ...$

s.t. $l(\theta^0) < l(\theta^1) < ... < l(\theta^t) < ....$

$$l(\theta)-l(\theta^t)=\log P(D\mid\theta)-\log P(D\mid\theta^t)$$

$$=\sum_{m=1}^{M}\log\sum_{y}P(D_m,y\mid\theta)-\sum_{m=1}^{M}\log\sum_{y}P(D_m,y\mid\theta^t)$$

$$=\sum_{m=1}^{M}\log\frac{\sum_{y}P(D_m,y\mid\theta)}{\sum_{y}P(D_m,y\mid\theta^t)}=\sum_{m=1}^{M}\log\sum_{y}\frac{P(D_m,y\mid\theta)}{\sum_{y'}P(D_m,y'\mid\theta^t)}$$

$$=\sum_{m=1}^{M}\log\sum_{y}\frac{P(D_m,y\mid\theta)}{\sum_{y'}P(D_m,y'\mid\theta^t)}\times\frac{P(D_m,y\mid\theta^t)}{P(D_m,y\mid\theta^t)}=\sum_{i=1}^{M}\log\sum_{y}\frac{P(D_m,y\mid\theta^t)}{\sum_{y'}P(D_m,y'\mid\theta^t)}\times\frac{P(D_m,y\mid\theta)}{P(D_m,y\mid\theta^t)}$$

$$=\sum_{m=1}^{M}\log\sum_{y}P(y\mid D_m,\theta^t)\frac{P(D_m,y\mid\theta)}{P(D_m,y\mid\theta^t)}=\sum_{m=1}^{M}\log E_{P(y|D_m,\theta^t)}[\frac{P(D_m,y\mid\theta)}{P(D_m,y\mid\theta^t)}]$$

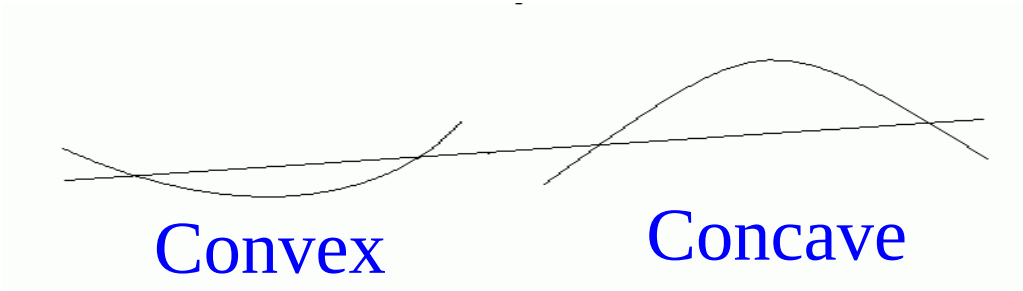
$$\geq\sum_{m=1}^{M}E_{P(y|D_m,\theta^t)}[\log\frac{P(D_m,y\mid\theta)}{P(D_m,y\mid\theta^t)}]$$

Jensen's inequality

Jensen's inequality

$$if \quad f \ is \cup -convex, then \ E[f(g(x)] \geq f(E[g(x])$$

$$if \quad f \ is \cap -concave, then \ E[f(g(x)] \leq f(E[g(x])$$



log is a concave function

$$E[\log(\ p(x)] \leq \log(\ E[\ p(x)])$$

Jensen's inequality corollary

- Let

$$\sum_j q_j = 1$$

$$q_j \geq 0$$

$$g(j) \geq 0$$

- Function log is concave, so from Jensen inequality we have:

$$\log(E[g]) \geq E[\log(g)]$$

$$\log(\sum_j q_j g(j)) \geq \sum_j q_j \log(g(j))$$

$$\log(\sum_j q_j g(j)) \geq \sum_j \log(g(j)^{q_j})$$

$$\log(\sum_j q_j g(j)) \geq \log\left(\prod_j g(j)^{q_j}\right)$$

$$\sum_j q_j g(j)) \geq \prod_j g(j)^{q_j} \qquad (1)$$

Lower bound lemma

$$If \quad \forall x: \ f(x) \le g(x)$$

$$\exists x_0: \ f(x_0) = g(x_0)$$

$$\exists \hat{x}: \ \hat{x} = \arg\max_{x} f(x)$$

Then $g(\hat{x}) \ge g(x_0)$

Proof: $g(\hat{x}) \ge f(\hat{x}) \ge f(x_0) = g(x_0)$

L(θ) is non-decreasing

$$l(\theta) - l(\theta^t) \geq \sum_{m=1}^{M} E_{P(y|D_m,\theta^t)}[\log \frac{P(D_m, y \mid \theta)}{P(D_m, y \mid \theta^t)}]$$

Let $g(\theta) = l(\theta) - l(\theta^t)$

$$f(\theta) = \sum_{m=1}^{M} E_{P(y|D_m,\theta^t)}[\log \frac{P(D_m, y \mid \theta)}{P(D_m, y \mid \theta^t)}]$$

So, $g(\theta) \geq f(\theta)$ *and* $g(\theta^t) = f(\theta^t) = 0$

$$\theta^{t+1} = \arg\max_{\theta} f(\theta)$$

$\Rightarrow$ $g(\theta^{t+1}) \geq g(\theta^t) = 0$ (By lower bound lemma)

$\Rightarrow$ $l(\theta^{t+1}) \geq l(\theta^t)$

Maximizing the lower bound

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_{m=1}^{M} E_{P(y|D_m,\theta^t)}[\log \frac{P(D_m, y \mid \theta)}{P(D_m, y \mid \theta^t)}]$$

$$= \arg\max_{\theta} \sum_{m=1}^{M} \sum_{y} P(y \mid D_m, \theta^t) \log \frac{P(D_m, y \mid \theta)}{P(D_m, y \mid \theta^t)}$$

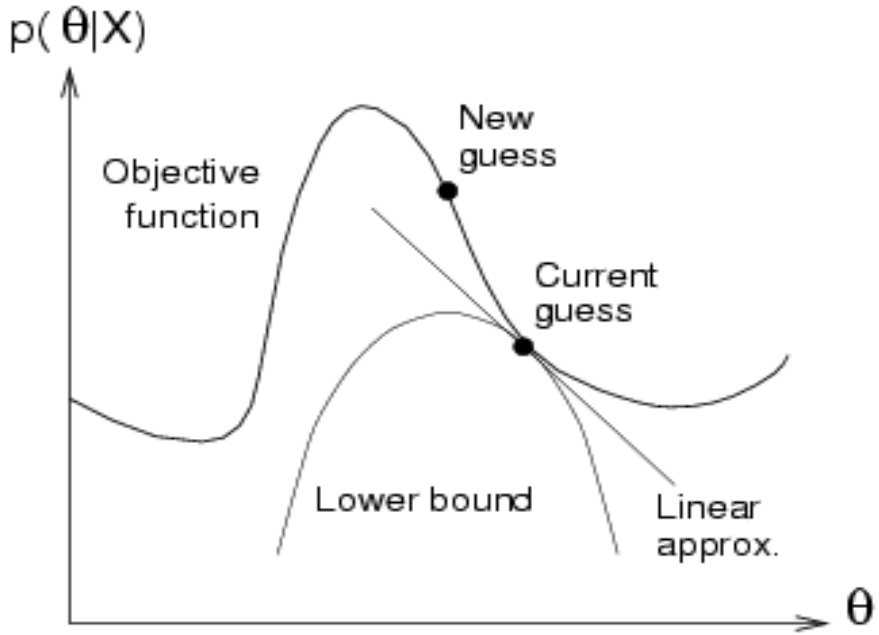
$$= \arg\max_{\theta} \sum_{m=1}^{M} \sum_{y} P(y \mid D_m, \theta^t) \log P(D_m, y \mid \theta)$$

$$= \arg\max_{\theta} \boxed{\sum_{m=1}^{M} E_{P(y|D_m,\theta^t)}[\log P(D_m, y \mid \theta)]}$$

The Q function

EM Algorithm

- EM is iterative technique designed for probabilistic models.



- Maximizing a function with lower-bound approximation vs. linear approximation

EM Algorithm

- EM makes a local approx. that is **lower bound** (l.b.) to the **objective function** (O.F.).
- Choosing a new guess to maximize the l.b. will always be an improvement, if gradient is not zero.
- Thus two steps: E – compute a l.b., M-maximize the l.b.

The Q-function

- Define the Q-function (a function of θ):

$$Q(\theta,\theta^t) = E[\log P(D,Y\mid\theta)\mid D,\theta^t] = E_{P(Y\mid D,\theta^t)}[\log P(D,Y\mid\theta)]$$

$$= \sum_Y P(Y\mid D,\theta^t)\log P(D,Y\mid\theta) = \sum_{m=1}^{M} E_{P(y\mid D_m,\theta^t)}[\log P(D_m,y\mid\theta)]$$

$$= \sum_{m=1}^{M}\sum_y P(y\mid D_m,\theta^t)\log P(D_m,y\mid\theta)$$

  - Y is a random vector.
  - D=(D_1, D_2, …, D_N) is a constant (vector).
  - Θ^t is the current parameter estimate and is a constant (vector).
  - Θ is the normal variable (vector) that we wish to adjust.
- The Q-function is the expected value of the complete data log-likelihood P(D,Y|θ) with respect to Y given D and θ^t.

The inner loop of the EM algorithm

- E-step: calculate

$$Q(\theta, \theta^t) = \sum_{m=1}^{M} \sum_{y} P(y \mid D_m, \theta^t) \log P(D_m, y \mid \theta)$$

- M-step: find $\theta^{(t+1)} = \arg \max_{\theta} Q(\theta, \theta^t)$
- L(θ) is non-decreasing at each iteration, the sequence:

$$\theta^0, \theta^1, ..., \theta^t, ...$$

- It can be proved that

$$l(\theta^0) < l(\theta^1) < ... < l(\theta^t) < ....$$

The inner loop of the Generalized EM algorithm (GEM)

- E-step: calculate

$$Q(\theta,\theta^t)=\sum_{m=1}^{M}\sum_{y}P(y\mid D_m,\theta^t)\log P(D_m,y\mid\theta)$$

- M-step: find

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

EM Algorithm:Coins

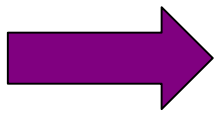
- Experiment
  - Two coins:
    - $P(\textit{Head on Coin 1}) = p$, $P(\textit{Head on Coin 2}) = q$
    - Experimenter first selects a coin: $P(\textit{Coin} = 1) = \alpha$
    - Chosen coin tossed 3 times (per experimental run)
  - Observe: $D = \{(1\ H\ H\ T), (1\ H\ T\ T), (2\ T\ H\ T)\}$
  - Want to predict: α, p, q
  - *How to model the problem?*
    - Now, can find most likely values of parameters α, p, q given data D

EM Algorithm:Coins

- Parameter Estimation
  - Fully observable case: easy to estimate p, q, and α
    - Suppose M heads are observed out of N coin flips
    - Maximum likelihood estimate(MLE) v_{ML} for $Flip_i$: $p = M/\mathrm{N}$

$$L(\theta) = \log P(D \mid \theta) = \log[\, p^M (1-p)^{N-M} \,]$$
$$= M \log p + (N-M)\log(1-p)$$

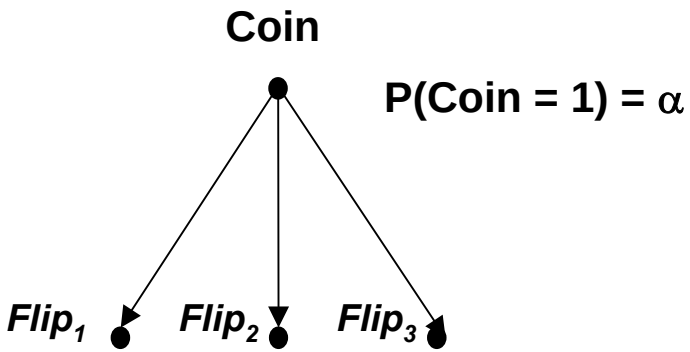
$$\frac{dL(\theta)}{dp} = \frac{d(M\log p + (N-M)\log(1-p))}{dp} = \frac{M}{p} - \frac{N-M}{1-p} = 0$$



p= M/N

EM Algorithm:Coins

- Parameter Estimation
  - Partially observable case
  - Don't know which coin the experimenter chose
  - Observe: D = {HHH,TTT,HTT,THH,HHT,TTH,HTH,THT}



$P(Flip_i = H \mid Coin = 1) = p$
$P(Flip_i = H \mid Coin = 2) = q$

EM Algorithm:Coins

- Problem
  - When we knew *Coin* = 1 or *Coin* = 2, there was no problem
  - *No known analytical solution* to the partially observable problem
    - i.e., not known how to compute estimates of p, q, and α to get v_{ML}
    - Moreover, not known what the computational complexity is

Solution:Iterative Estimation

- Given: an initial *guess* of $P(Coin = 1 \mid D)$, $P(Coin = 2 \mid D)$
- Generate "fictional data points", with probability weight:
  - $P(Coin = 1 \mid D) = P(D \mid Coin = 1)\, P(Coin = 1) / P(\mathrm{D})$ based on our guess of α, p, q
  - Expectation step (the "E" in EM)
- Find most likely values of parameters α, p, q given "fictional" data
  - Maximization step (the "M" in EM)
- Repeat until termination condition met.

Expectation Step

- Suppose we observed *m* actual experiments(data points), each *n* coin flips long
  - Each experiment corresponds to one choice of coin (α)
  - Let h_i denote the number of heads in experiment D_i (a single data point)
- Q: How did we simulate the "fictional" data points, E[$\sum$ log $P(D \mid \hat{\alpha}, \hat{p}, \hat{q})$]?
- A: By estimating (for $1 \le i \le m$, i.e., the real data points)

$$P_1^i = P(Coin = 1 \mid D_i) = \frac{P(D_i \mid Coin = 1) \cdot P(Coin = 1)}{P(D_i)}$$

$$= \frac{\hat{\alpha} \cdot \hat{p}^{h_i} (1 - \hat{p})^{n-h_i}}{\hat{\alpha} \cdot \hat{p}^{h_i} (1 - \hat{p})^{n-h_i} + (1 - \hat{\alpha}) \cdot \hat{q}^{h_i} (1 - \hat{q})^{n-h_i}}$$

Expectation Step

- So, the Expectation is:

$$E\left(\sum_i \log P(D_i \mid \hat{\alpha}, \hat{p}, \hat{q})\right)$$

$$= \sum_i P_1^i \log P(Coin = 1, D^i \mid \hat{\alpha}, \hat{p}, \hat{q}) + (1 - P_1^i) \log P(Coin = 2, D^i \mid \hat{\alpha}, \hat{p}, \hat{q})$$

$$= \sum_i P_1^i \log(\hat{\alpha}\, \hat{p}^{h_i} (1-\hat{p})^{n-h_i}) + (1 - P_1^i) \log((1-\hat{\alpha})\, \hat{q}^{h_i} (1-\hat{q})^{n-h_i})$$

$$= \sum_i P_1^i (\log \hat{\alpha} + h_i \log \hat{p} + (n - h_i) \log(1-\hat{p})) +$$

$$(1 - P_1^i)(\log(1-\hat{\alpha}) + h_i \log \hat{q} + (n - h_i) \log(1-\hat{q}))$$

Maximization Step

- Q: What are we updating? What objective function are we maximizing?
- A: We are updating $\hat{\alpha}, \hat{p}, \hat{q}$ to maximize $\frac{\partial E}{\partial \hat{\alpha}}, \frac{\partial E}{\partial \hat{p}}, \frac{\partial E}{\partial \hat{q}}$ where $E = \mathrm{E}\left[\sum_{i=1}^{m} \log P(D_i \mid \hat{\alpha}, \hat{p}, \hat{q})\right]$

$$\hat{\alpha} = \frac{\sum P(Coin = 1 \mid D_i)}{m}, \hat{p} = \frac{\sum \frac{h_i}{n} P(Coin = 1 \mid D_i)}{\sum P(Coin = 1 \mid D_i)}, \hat{q} = \frac{\sum \frac{h_i}{n} \left[1 - P(Coin = 1 \mid D_i)\right]}{\sum \left[1 - P(Coin = 1 \mid D_i)\right]}$$

Coins

- Y={Coin_1,Coin_2};
  D={HHH,TTT,HTT,THH,HHT,TTH,HTH,THT}
  θ={α, *p*, *q*}
- and
  P(D,Y | θ) = P(Y | θ)P(D |Y,θ)

Where,

$$P(Y \mid \theta) = \begin{cases} \alpha & \text{Y=Coin\_1;} \\ 1\text{-}\alpha & \text{Y=Coin\_2} \end{cases}$$

and

$$P(D \mid Y, \theta) = \begin{cases} p^h(1\text{-p})^t & \text{Y=Coin\_1;} \\ q^h(1\text{-q})^t & \text{Y=Coin\_2} \end{cases}$$

h:the number of head in D, t: the number of tail in D

Coins

- Various probabilities can be calculated, for example:

$$P(THT, Coin\_1 \mid \theta) = \mathrm{P}(Coin\_1 \mid \theta)\mathrm{P}(THT \mid Coin\_1, \theta) = \alpha p(1-p)^2$$

$$P(THT, Coin\_2 \mid \theta) = (1-\alpha)q(1-q)^2$$

$$P(THT \mid \theta) = P(THT, Coin\_1 \mid \theta) + P(THT, Coin\_2 \mid \theta)$$

$$= \alpha p(1-p)^2 + (1-\alpha)q(1-q)^2$$

$$\Rightarrow$$

$$P(Coin\_1 \mid THT, \theta) = \frac{P(THT, Coin\_1 \mid \theta)}{P(THT \mid \theta)}$$

$$= \frac{\alpha p(1-p)^2}{\alpha p(1-p)^2 + (1-\alpha)q(1-q)^2}$$

Coins

- Partially observed data might look like:
- <HHH>; <TTT>; <HHH>; <TTT>; <HHH>
- If current parameters are $\theta=\{\alpha, p, q\}$

$$P(Coin\_1 \mid HHH, \theta) = \frac{P(HHH, Coin\_1 \mid \theta)}{P(HHH \mid \theta)}$$

$$= \frac{\alpha p^3}{\alpha p^3 + (1-\alpha) q^3}$$

$$P(Coin\_1 \mid TTT, \theta) = \frac{P(TTT, Coin\_1 \mid \theta)}{P(TTT \mid \theta)}$$

$$= \frac{\alpha (1-p)^3}{\alpha (1-p)^3 + (1-\alpha)(1-q)^3}$$

Coins

- If current parameters are $\theta=\{\alpha, p, q\}$

$$P(Coin\_1|HHH,\theta)=\frac{P(HHH,Coin\_1|\theta)}{P(HHH|\theta)}=\frac{\alpha p^3}{\alpha p^3+(1-\alpha)q^3}$$

$$P(Coin\_1|TTT,\theta)=\frac{P(TTT,Coin\_1|\theta)}{P(TTT|\theta)}=\frac{\alpha(1-p)^3}{\alpha(1-p)^3+(1-\alpha)(1-q)^3}$$

$$if \quad \alpha=.3, \quad p=.3, \quad q=.6$$

$$P(Coin\_1|HHH,\theta)=.0508$$

$$P(Coin\_1|TTT,\theta)=.6967$$

Coins

- After filling in hidden variables for each example, partially observed data might look like:

(HHH,Coin_1)	P(Coin_1|HHH)=.0508
(HHH,Coin_2)	P(Coin_2|HHH)=.9492
(TTT,Coin_1)	P(Coin_1|TTT)=.6967
(TTT,Coin_2)	P(Coin_2|TTT)=.3033
(HHH,Coin_1)	P(Coin_1|HHH)=.0508
(HHH,Coin_2)	P(Coin_2|HHH)=.9492
(TTT,Coin_1)	P(Coin_1|TTT)=.6967
(TTT,Coin_2)	P(Coin_2|TTT)=.3033
(HHH,Coin_1)	P(Coin_1|HHH)=.0508
(HHH,Coin_2)	P(Coin_2|HHH)=.9492

Coins

- New Estimates:

(HHH,Coin_1) P(Coin_1|HHH)=.0508

(HHH,Coin_2) P(Coin_2|HHH)=.9492

(TTT,Coin_1) P(Coin_1|TTT)=.6967

(TTT,Coin_2) P(Coin_2|TTT)=.3033

$$\hat{\alpha}=\frac{\sum P(Coin=1|D_i)}{m},\ \hat{p}=\frac{\sum \frac{h_i}{n} P(Coin=1|D_i)}{\sum P(Coin=1|D_i)},\ \hat{q}=\frac{\sum \frac{h_i}{n}\left[1-P(Coin=1|D_i)\right]}{\sum\left[1-P(Coin=1|D_i)\right]}$$

$$\alpha=\frac{.0508\times 3+.6967\times 2}{5}=.3092$$

$$p=\frac{3\times 3\times .0508+0\times 2\times .6967}{3\times 3\times .0508+3\times 2\times .6967}=.0987$$

$$q=\frac{3\times 3\times .9492+0\times 2\times .3033}{3\times 3\times .9492+3\times 2\times .3033}=.8244$$

Coins

- Begin with parameters α=.3; p=.3; q=.6
- Fill in losing variables, using

$$P(Coin\_1 | HHH, \theta) = .0508$$
$$P(Coin\_1 | TTT, \theta) = .6967$$

- Re-estimate parameters to be

$$\alpha = .3092; \quad p = .0987; \quad q = .8244$$

Problems with EM

- Only local optimum.
- If priors are uniform, may be impossible to make any progress…
- … next figure illustrates the need for some randomization to move us off an uninformative prior…

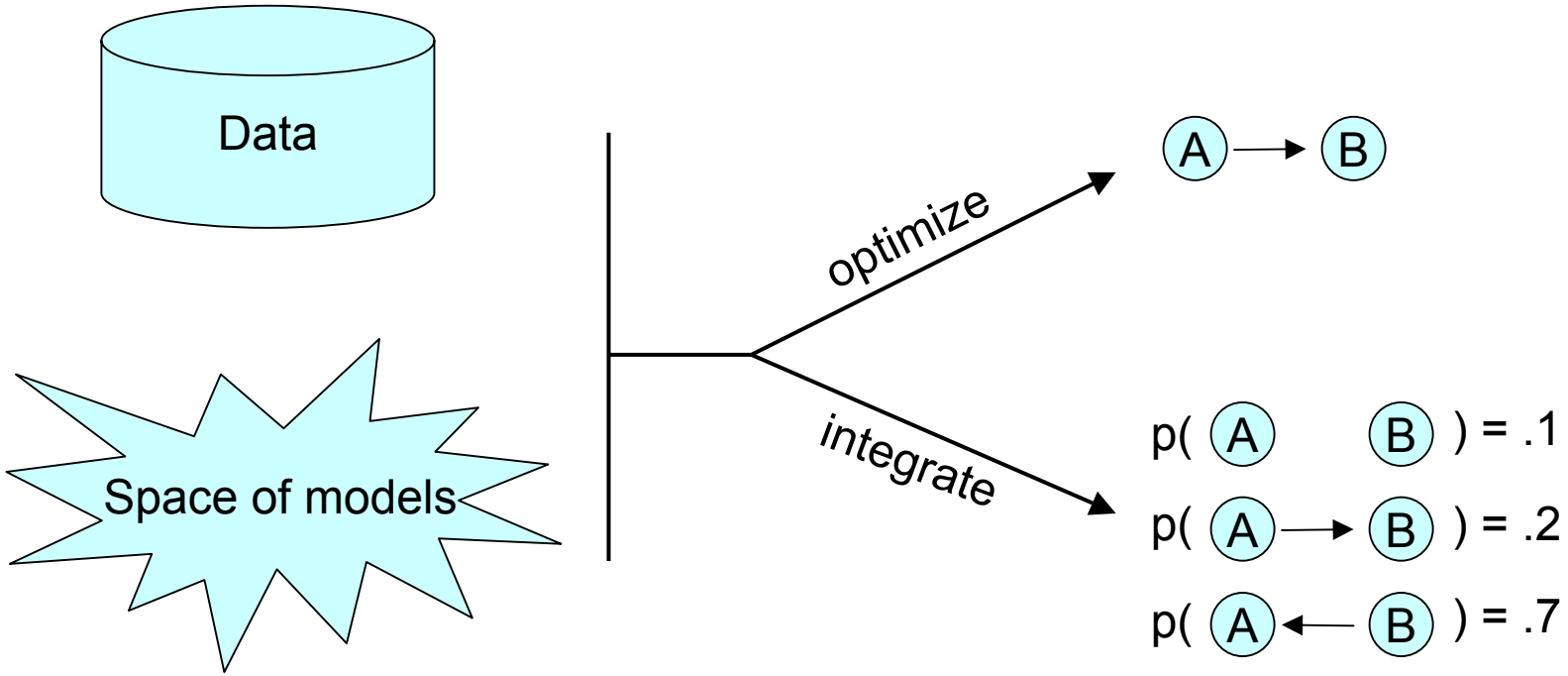
Outline

- Overview
- Parameter Learning with full data
- Parameter Learning with partial data
- **Structure+Parameter Learning with full data**
- Learning parameter of MRF

Unknown structure+ full observability

- **Model selection**
- **BN has two components**:
  - **Structure of the network** (models conditional independences)
  - **A set of parameters** (conditional child-parent distributions)
- **Assumption:**
  - All variables are observable in the training set
- **Question**
  - how to **learn the structure** of the BN?

Unknown structure+ full observability

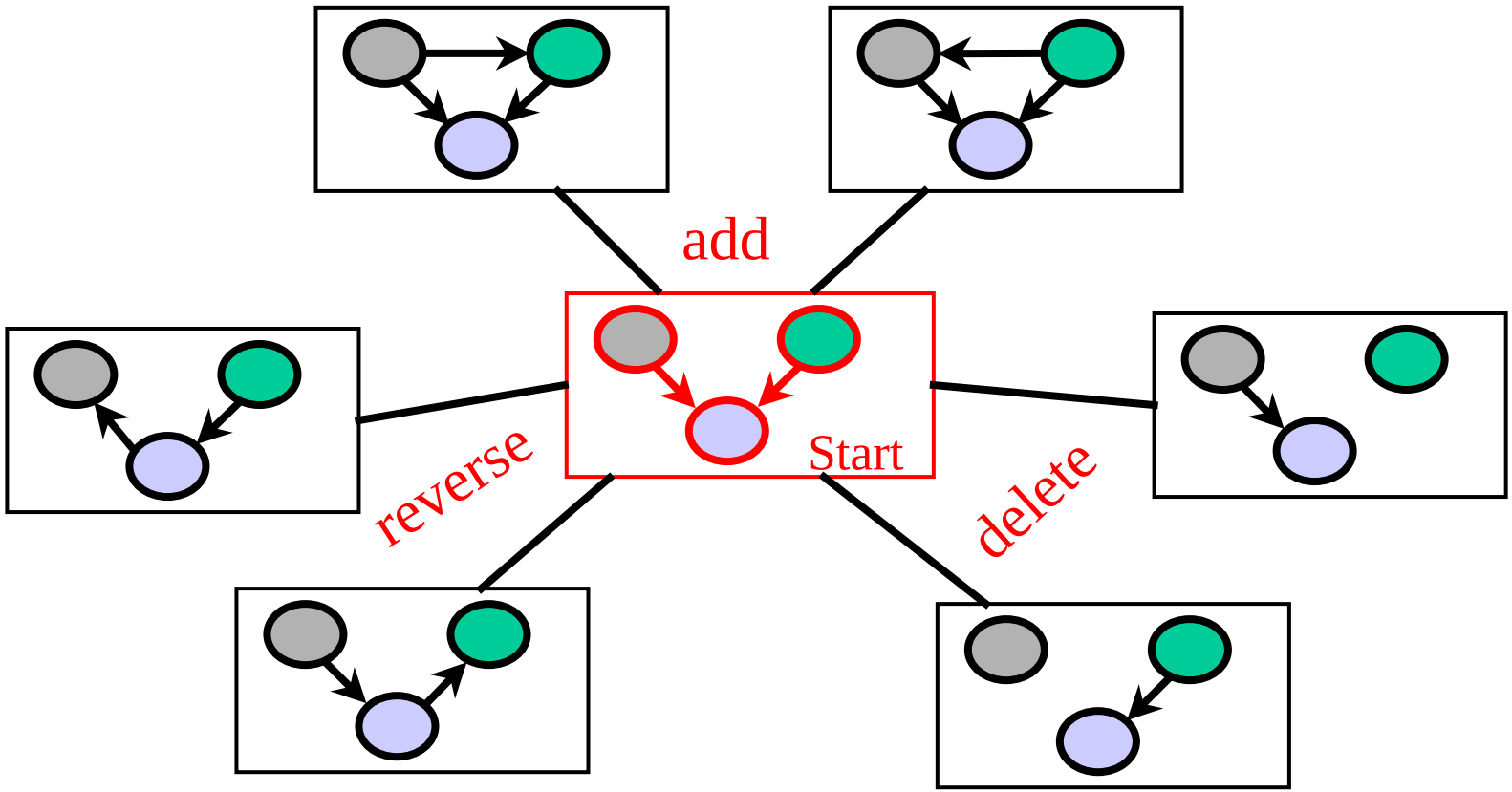


- Model space = DAGs with CPD for each node

Search space

Space = network structures

Operators = **add**/**reverse**/**delete edges**



Complexity

Model selection: find the best ***Structure*** for all possible S

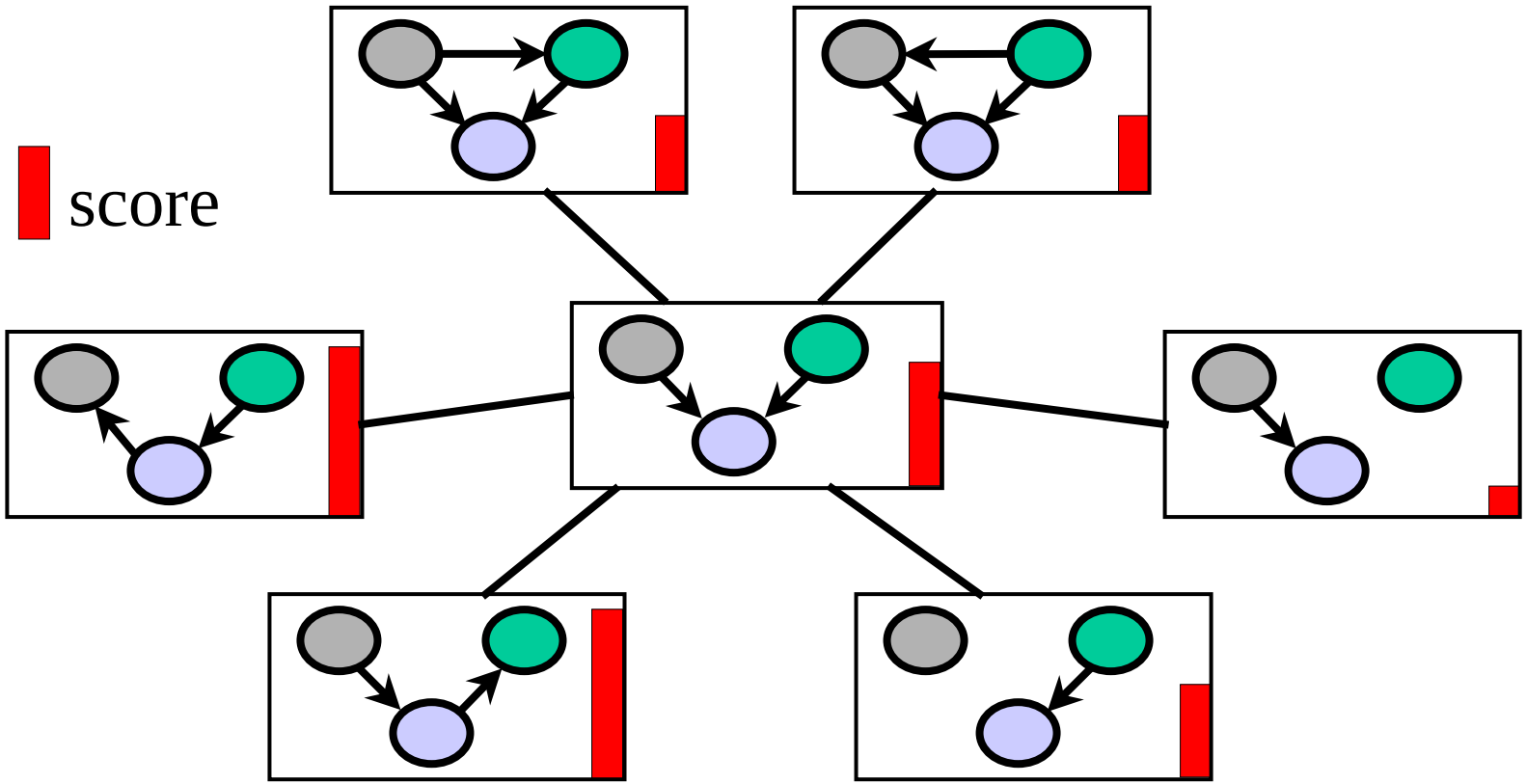
- Number of DAGs is superexponential in # of nodes
- 4 node DAGs...
  - 543 possible S
  - …

Question: How to search the best structure in the huge amount of possible DAGs?

n	S(n)
1	1
2	3
3	25
4	543
5	29,281
6	3,781,503
7	1.1e9
...	...
n	

Heuristic search

Use **scoring function** to do heuristic search (any algorithm).
Greedy hill-climbing.



Outline

- Overview
- Parameter Learning with full data
- Parameter Learning with partial data
- Structure+Parameter Learning with full data
- **Learning Parameter of MRF**