# Lecture 23: April 15, 2015
# cs 573: Probabilistic Reasoning
# Professor Nevatia
# Spring 2015

# Review

- HW#7 assigned; due April 22
- Exam2: April 29, class period, here
  - Closed book/notes
  - Detailed list of topics to be posted
- Previous Lecture
  - Bayesian Parameter learning
    - Incorporates effect of priors, less sensitive with small training data
    - Conjugate priors (beta/Dirichlet functions)
  - Intro to case of learning from incomplete data
- Today's objective
  - Learning BNs from incomplete data

# Likelihood Function with Missing Data

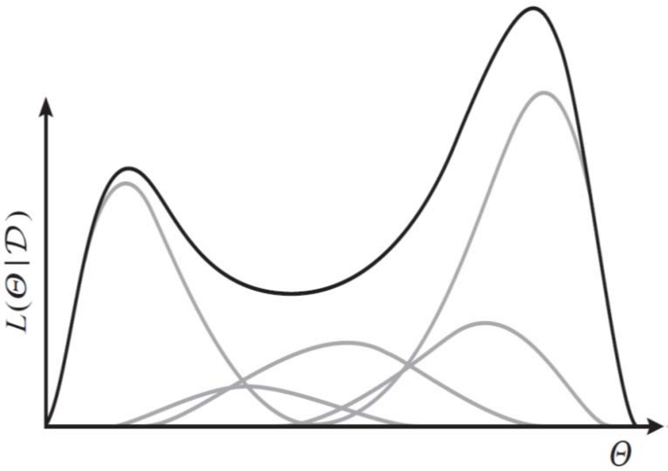
- Consider a network G with set of variables **X**
- At mth instance, let **O**[m] be the observed variables with values **o**[m]; **H**[m] be the set of missing or hidden variables
- $L(\theta : D) = P(D| \theta) = \prod_{m=1,M} P(o[m] \mid \theta)$

  $= \prod_{m=1,M} \sum_{h[m]} P(o[m],h[m] : \theta)$ for *iid* samples
- Consider a chain A → B → C; let B be "hidden"
  - Suppose we observe $a^0$ and $c^0$
  - $P(a^0,c^0) = \sum_B P(a^0,B,c^0)$

    $= P(a^0)\ \{P(b^0|a^0)\ P(c^0|b^0) + P(b^1|a^0)\ P(c^0|b^1)\}$
  - Note: dependent on P(B|A) and P (C|B)
  - Probability of *k* such samples is $P(a^0,c^0)^k$
  - likelihood function over all samples is a product of such terms for various assignments of A and C
    - Each term in product, however, is some of other likelihood functions, so when we take a log, terms do not separate out

# Properties of Likelihood Function

- Even though each term in the sum is *log-concave* (unimodal) their sum is not so, hence optimization (to find MLE) is difficult



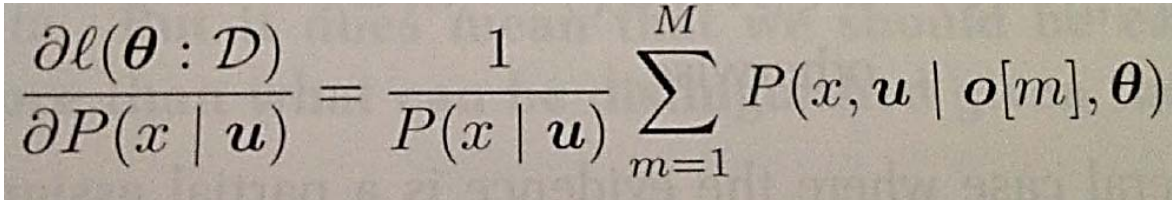
- Must sum over joint assignments to all unobserved variables => exponential (in number of hidden nodes) number of sums
- Even computation of the likelihood of a sample is complex (requires an inference on the network)
- Need to use optimization methods to maximize likelihood
  - Gradient Ascent method
  - Expectation Maximization (EM) algorithm

# Identifiability

- Many solutions may be equivalent; under-constrained problem
  - Likelihood function has a flat top
  - In our simple example, many combinations of P(B|A) and P(C|B) may give the same values for $P(a^0,c^0)$
- Another Example:
  - Two types of tacks that are tossed
  - Have different probability of coming up “heads”
  - Tacks got mixed so don’t know which outcome is from which tack; probability of selecting one is not the same as other
    - We want to estimate this probability also
  - This is still MAR condition but many choices of distributions of choice of tacks and probability of heads can give same likelihood of observed data.

# Gradient Calculation for BNs

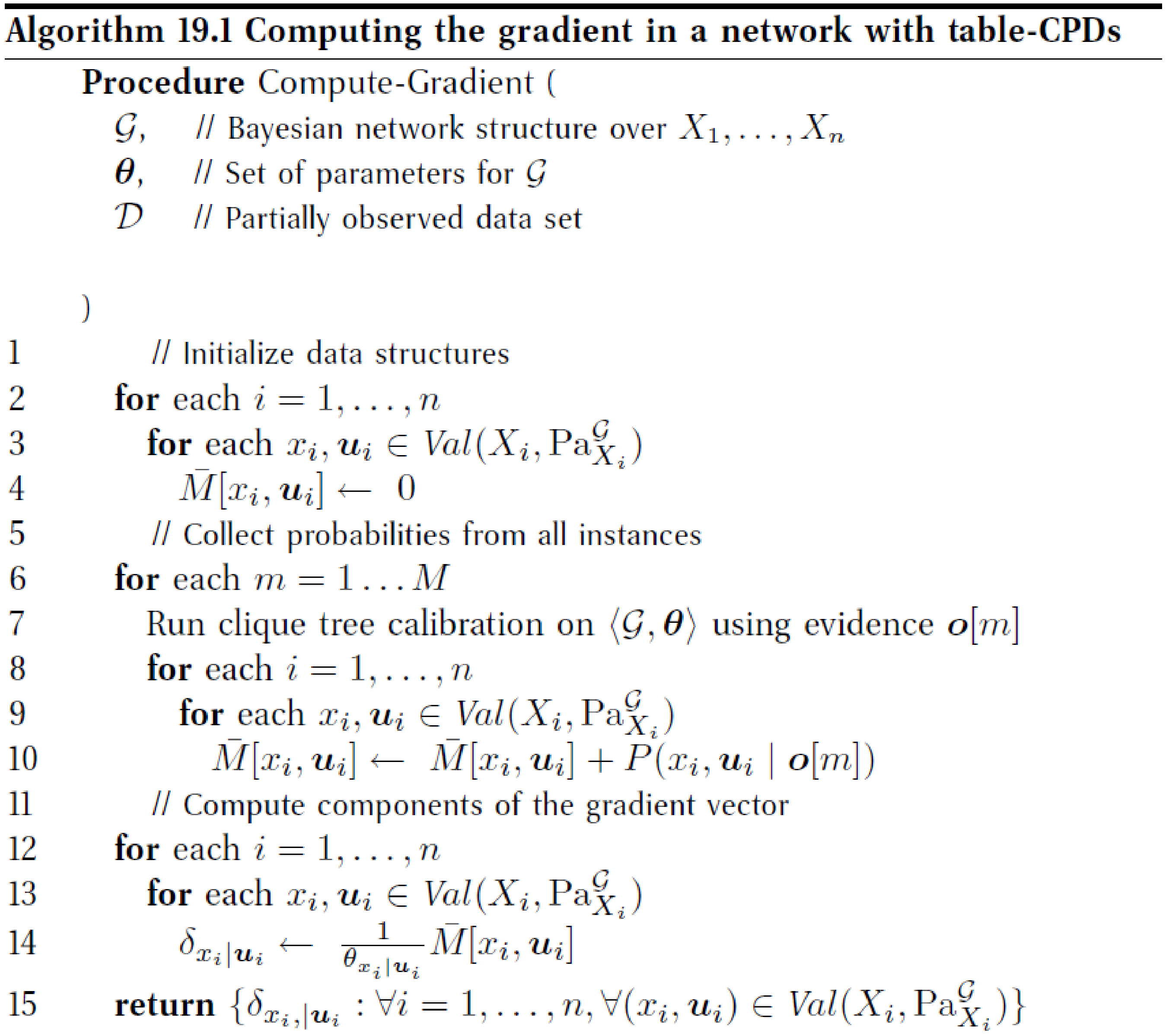
- Consider a BN where X is a child node with a set of parents **U**; Let **o** be a tuple of observations (assignments of some variables)
- let D = {**o**[1],… **o**[M]}, a set of observations with possibly different missing variables for each sample
- Can be shown that:



$$\frac{\partial \ell(\boldsymbol{\theta} : \mathcal{D})}{\partial P(x \mid \boldsymbol{u})} = \frac{1}{P(x \mid \boldsymbol{u})} \sum_{m=1}^{M} P(x, \boldsymbol{u} \mid \boldsymbol{o}[m], \boldsymbol{\theta})$$

  - Note that derivative wrt each term in the distribution can be computed independently (together they define the gradient)
  - Contribution from multiple samples is summed together
  - The term inside the sum requires making an inference on the entire set of variables, once for each different sample
  - Book provides a numerical example; tedious to cover in class

# Algorithm 19.1, Computing the gradient



**Algorithm 19.1 Computing the gradient in a network with table-CPDs**

```
Procedure Compute-Gradient (
    $\mathcal{G}$,   // Bayesian network structure over $X_1, \ldots, X_n$
    $\boldsymbol{\theta}$,   // Set of parameters for $\mathcal{G}$
    $\mathcal{D}$    // Partially observed data set
)
    // Initialize data structures
    for each $i = 1, \ldots, n$
        for each $x_i, \boldsymbol{u}_i \in Val(X_i, \mathrm{Pa}^{\mathcal{G}}_{X_i})$
            $\bar{M}[x_i, \boldsymbol{u}_i] \leftarrow 0$
    // Collect probabilities from all instances
    for each $m = 1 \ldots M$
        Run clique tree calibration on $\langle \mathcal{G}, \boldsymbol{\theta} \rangle$ using evidence $\boldsymbol{o}[m]$
        for each $i = 1, \ldots, n$
            for each $x_i, \boldsymbol{u}_i \in Val(X_i, \mathrm{Pa}^{\mathcal{G}}_{X_i})$
                $\bar{M}[x_i, \boldsymbol{u}_i] \leftarrow \bar{M}[x_i, \boldsymbol{u}_i] + P(x_i, \boldsymbol{u}_i \mid \boldsymbol{o}[m])$
    // Compute components of the gradient vector
    for each $i = 1, \ldots, n$
        for each $x_i, \boldsymbol{u}_i \in Val(X_i, \mathrm{Pa}^{\mathcal{G}}_{X_i})$
            $\delta_{x_i|\boldsymbol{u}_i} \leftarrow \frac{1}{\theta_{x_i|\boldsymbol{u}_i}} \bar{M}[x_i, \boldsymbol{u}_i]$
    return $\{\delta_{x_i,|\boldsymbol{u}_i} : \forall i = 1, \ldots, n, \forall (x_i, \boldsymbol{u}_i) \in Val(X_i, \mathrm{Pa}^{\mathcal{G}}_{X_i})\}$
```

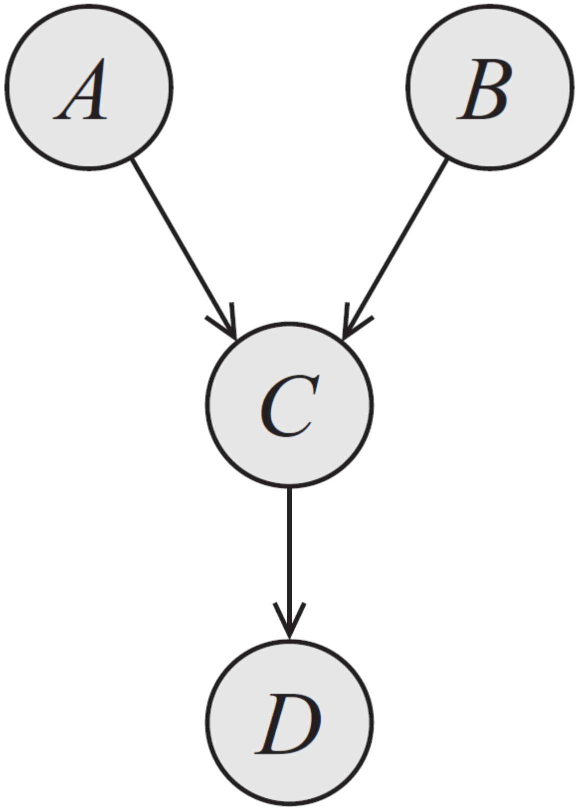
# Gradient Ascent

- As we have a method to compute gradient, given the samples and current estimate of parameters, we can use gradient ascent to maximize
- It is easy to see that the partial derivatives are always positive (sum of probabilities)
  - Can not just increase all as terms in one CPD must add to one; increasing one requires decreasing others
- Use a constrained optimization method (Lagrange multipliers)
- Book provides a reparameterization that preserves a legal distribution so unconstrained optimization may be used
- Note that the function has many maxima (exponential in number of unobserved variables) so finding global maximum may be difficult
  - Try multiple initializations, random perturbations…

# Expectation Maximization (EM) Algorithm

- If we have complete data, it is easy to estimate the parameters (compute frequencies)
- If we know the parameters, we can infer the distribution over values of the hidden variables
- "*Chicken and egg*" problem
- Start with an initial parameter assignment, say $\theta^0$ .
- Use $\theta^0$ to compute posterior distribution over possible assignments to hidden variables (as in gradient ascent)
- Use these assignments to compute new parameters, say $\theta^1$, that maximizes the expected likelihood
- Iterate on the two steps above.
- It can be shown that in each iteration, likelihood necessarily increases or stays constant. Thus, the procedure converges to a local maximum.
- We assume that values are "missing at random", not as a function of other *unobserved* values.

# EM Example



For fully observable case

$$\hat{\theta}_{d^1|c^0} = \frac{M[d^1, c^0]}{M[c^0]} = \frac{\sum_{m=1}^{M} \mathbf{1}\{\xi[m]\langle D, C\rangle = \langle d^1, c^0\rangle\}}{\sum_{m=1}^{M} \mathbf{1}\{\xi[m]\langle C\rangle = c^0\}}.$$

# With Hidden Variables

- Let o = <$a^1$, ?, ?, $d^0$>
- Four possible ways to fill in the two missing variable values: <$b^1$, $c^1$>, <$b^1$, $c^0$>, <$b^0$, $c^1$>, <$b^0$, $c^0$>
- Let initial values of parameters **θ** be as follows:

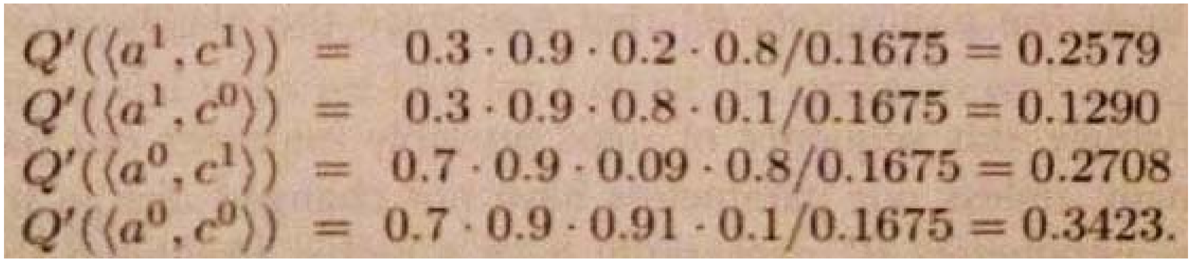
$$\begin{array}{llll} \theta_{a^1} & = 0.3 & \theta_{b^1} & = 0.9 \\ \theta_{d^1|c^0} & = 0.1 & \theta_{d^1|c^1} & = 0.8 \\ \theta_{c^1|a^0,b^0} & = 0.83 & \theta_{c^1|a^1,b^0} & = 0.6 \\ \theta_{c^1|a^0,b^1} & = 0.09 & \theta_{c^1|a^1,b^1} & = 0.2, \end{array}$$

- Let Q(B,C) = P(B, C | $a^1$, $d^0$, **θ**)

$$\begin{aligned} Q(\langle b^1, c^1\rangle) &= 0.3 \cdot 0.9 \cdot 0.2 \cdot 0.2/0.2196 = 0.0492 \\ Q(\langle b^1, c^0\rangle) &= 0.3 \cdot 0.9 \cdot 0.8 \cdot 0.9/0.2196 = 0.8852 \\ Q(\langle b^0, c^1\rangle) &= 0.3 \cdot 0.1 \cdot 0.6 \cdot 0.2/0.2196 = 0.0164 \\ Q(\langle b^0, c^0\rangle) &= 0.3 \cdot 0.1 \cdot 0.4 \cdot 0.9/0.2196 = 0.0492, \end{aligned}$$

Numerator is product of appropriate probabilities Denominator is the normalizing constant =P($a^1$,$d^0$ )

- For example of < ?, $b^1$, ?, $d^1$>



$$\begin{aligned} Q'(\langle a^1, c^1\rangle) &= 0.3 \cdot 0.9 \cdot 0.2 \cdot 0.8/0.1675 = 0.2579 \\ Q'(\langle a^1, c^0\rangle) &= 0.3 \cdot 0.9 \cdot 0.8 \cdot 0.1/0.1675 = 0.1290 \\ Q'(\langle a^0, c^1\rangle) &= 0.7 \cdot 0.9 \cdot 0.09 \cdot 0.8/0.1675 = 0.2708 \\ Q'(\langle a^0, c^0\rangle) &= 0.7 \cdot 0.9 \cdot 0.91 \cdot 0.1/0.1675 = 0.3423. \end{aligned}$$

# Compute New Parameters

- Take the *completions* to provide fully observable samples
- In general, let **H**[m] denote the variables hidden in instance **o**[m]
- Construct a new data set $D^+$ consisting of
- $U_m$ {<**o**[m], **h**[m]> : **h**[m] is in val(**H**[m]) }
- Each data case <**o**[m], **h**[m]> has weight Q (**h**[m]) = P (**h**[m], **o**[m], **θ** )
- Computed *expected* sufficient statistics (ESS)

$$\bar{M}_{\boldsymbol{\theta}}[\boldsymbol{y}] = \sum_{m=1}^{M} \sum_{\boldsymbol{h}[m] \in Val(\boldsymbol{H}[m])} Q(\boldsymbol{h}[m]) \boldsymbol{I}\{\xi[m]\langle \boldsymbol{Y} \rangle = \boldsymbol{y}\}..$$

- Use ESS to compute probabilities as in usual MLE formula
- Example on next slide

# Example

- Use ESS to compute probabilities as in usual MLE formula

$$\tilde{\theta}_{d^1|c^0} = \frac{\bar{M}_{\theta}[d^1, c^0]}{\bar{M}_{\theta}[c^0]}.$$

$$\begin{aligned} \bar{M}_{\theta}[d^1, c^0] &= Q'(\langle a^1, c^0 \rangle) + Q'(\langle a^0, c^0 \rangle) \\ &= 0.1290 + 0.3423 = 0.4713 \end{aligned}$$

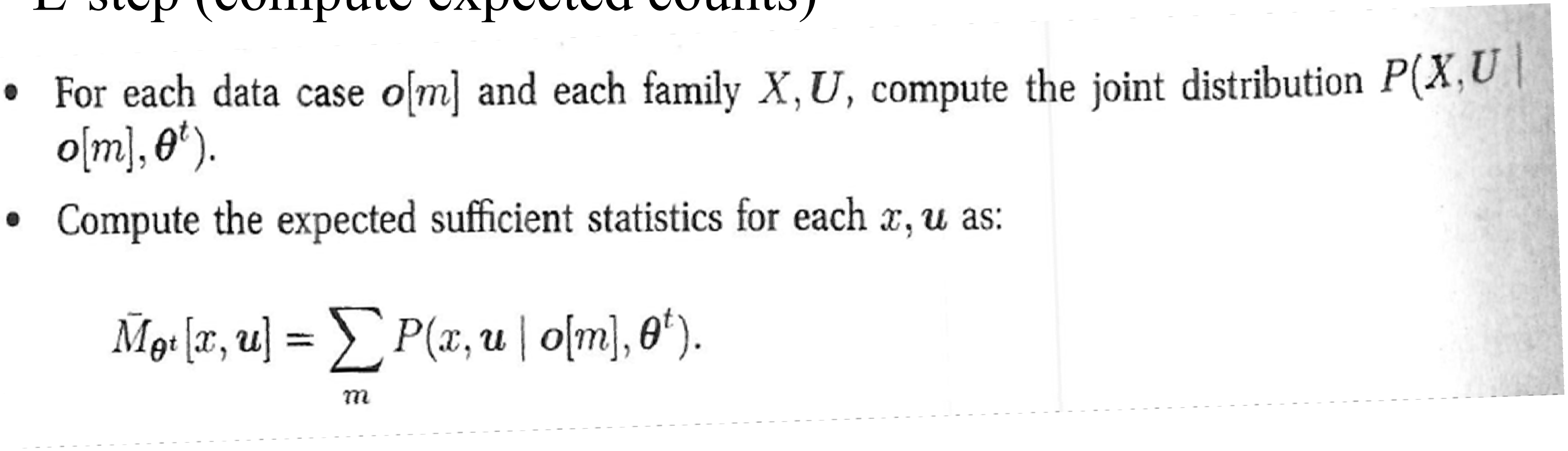
$$\begin{aligned} \bar{M}_{\theta}[c^0] &= Q(\langle b^1, c^0 \rangle) + Q(\langle b^0, c^0 \rangle) + Q'(\langle a^1, c^0 \rangle) + Q'(\langle a^0, c^0 \rangle) \\ &= 0.8852 + 0.0492 + 0.1290 + 0.3423 = 1.4057. \end{aligned}$$

$$\tilde{\theta}_{d^1|c^0} = \frac{0.4713}{1.4057} = 0.3353.$$

- Note: this procedure is not feasible, in general, as the number of completions is exponential in the number of hidden variables
  - Fortunately, such enumeration is not necessary, see next slide

# EM for Bayesian Networks (Table CPDs)

- E-step (compute expected counts)



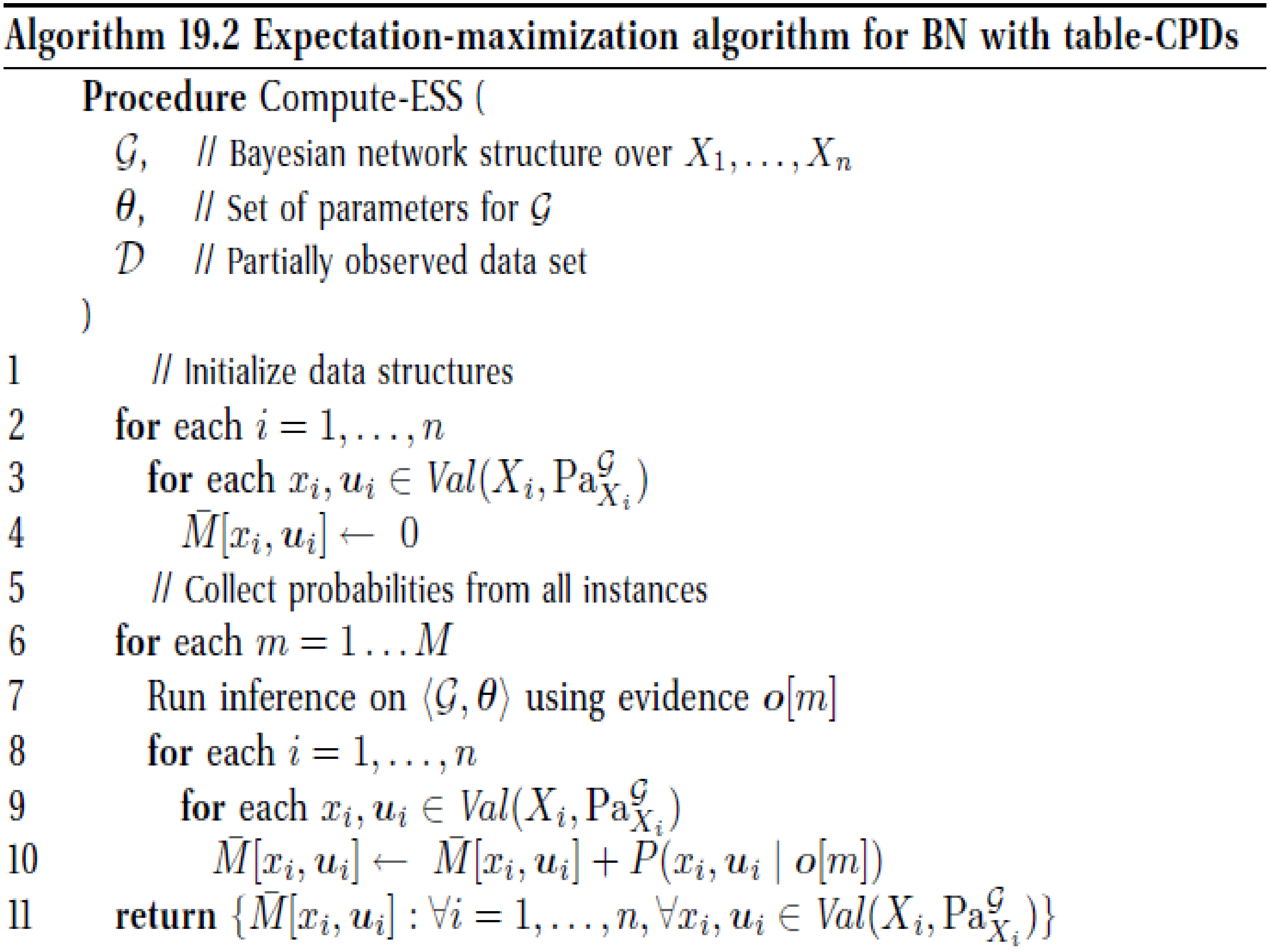
- For each data case $\boldsymbol{o}[m]$ and each family $X, \boldsymbol{U}$, compute the joint distribution $P(X, \boldsymbol{U} \mid \boldsymbol{o}[m], \boldsymbol{\theta}^t)$.
- Compute the expected sufficient statistics for each $x, \boldsymbol{u}$ as:

$$\bar{M}_{\boldsymbol{\theta}^t}[x, \boldsymbol{u}] = \sum_m P(x, \boldsymbol{u} \mid \boldsymbol{o}[m], \boldsymbol{\theta}^t).$$

- Note that (x, u) will occur in the same clique in a clique-tree
- M-Step (adjust parameters to achieve MLE)

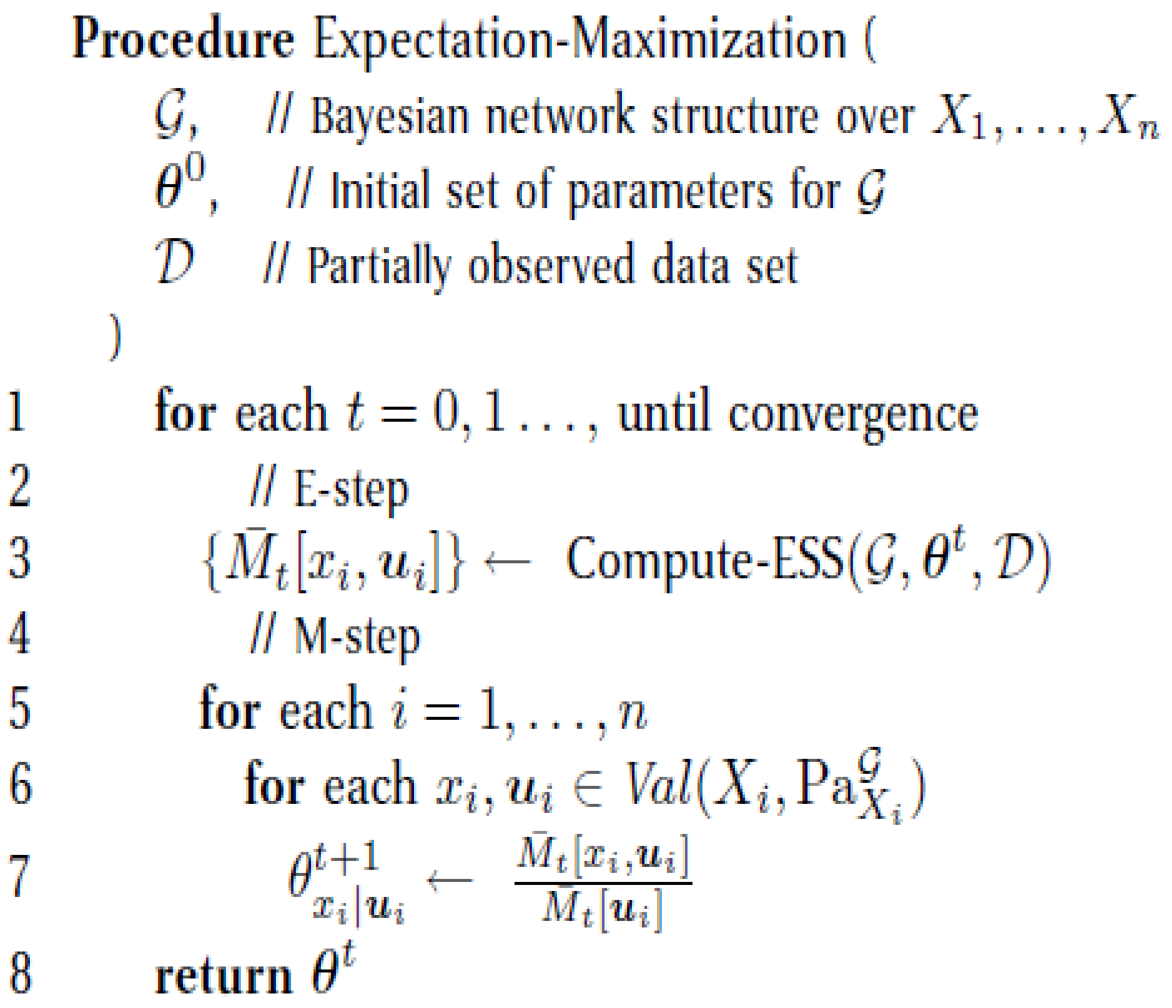
$$\theta^{t+1}_{x|\boldsymbol{u}} = \frac{\bar{M}_{\boldsymbol{\theta}^t}[x, \boldsymbol{u}]}{\bar{M}_{\boldsymbol{\theta}^t}[\boldsymbol{u}]}.$$

- Algorithm 19.2
- Note the method requires inferences at each iteration for each data sample
- We have not shown that the above M-step does maximize the likelihood function



**Algorithm 19.2 Expectation-maximization algorithm for BN with table-CPDs**

```
Procedure Compute-ESS (
    $\mathcal{G}$,    // Bayesian network structure over $X_1, \ldots, X_n$
    $\theta$,    // Set of parameters for $\mathcal{G}$
    $\mathcal{D}$    // Partially observed data set
)
    // Initialize data structures
    for each $i = 1, \ldots, n$
        for each $x_i, \boldsymbol{u}_i \in Val(X_i, \mathrm{Pa}^{\mathcal{G}}_{X_i})$
            $\bar{M}[x_i, \boldsymbol{u}_i] \leftarrow 0$
    // Collect probabilities from all instances
    for each $m = 1 \ldots M$
        Run inference on $\langle \mathcal{G}, \theta \rangle$ using evidence $\boldsymbol{o}[m]$
        for each $i = 1, \ldots, n$
            for each $x_i, \boldsymbol{u}_i \in Val(X_i, \mathrm{Pa}^{\mathcal{G}}_{X_i})$
                $\bar{M}[x_i, \boldsymbol{u}_i] \leftarrow \bar{M}[x_i, \boldsymbol{u}_i] + P(x_i, \boldsymbol{u}_i \mid \boldsymbol{o}[m])$
    return $\{\bar{M}[x_i, \boldsymbol{u}_i] : \forall i = 1, \ldots, n, \forall x_i, \boldsymbol{u}_i \in Val(X_i, \mathrm{Pa}^{\mathcal{G}}_{X_i})\}$
```



```
Procedure Expectation-Maximization (
    $\mathcal{G}$,    // Bayesian network structure over $X_1, \ldots, X_n$
    $\theta^0$,    // Initial set of parameters for $\mathcal{G}$
    $\mathcal{D}$    // Partially observed data set
  )
    for each $t = 0, 1 \ldots,$ until convergence
        // E-step
      $\{\bar{M}_t[x_i, \boldsymbol{u}_i]\} \leftarrow$ Compute-ESS($\mathcal{G}, \theta^t, \mathcal{D}$)
        // M-step
      for each $i = 1, \ldots, n$
        for each $x_i, \boldsymbol{u}_i \in Val(X_i, \mathrm{Pa}^{\mathcal{G}}_{X_i})$
          $\theta^{t+1}_{x_i|\boldsymbol{u}_i} \leftarrow \frac{\bar{M}_t[x_i,\boldsymbol{u}_i]}{\bar{M}_t[\boldsymbol{u}_i]}$
    return $\theta^t$
```

# Comments on EM for BNs

- Can be shown that each iteration of EM increases the likelihood
- Thus, EM converges to a stationary point of the likelihood function
- Can be shown that the stationary point is a local maximum in "almost all" cases
- However, many local maxima may exist
- How to find global maximum?
  - Usual methods such as:
  - use prior domain knowledge
  - start from multiple initial positions
  - perturb the solutions randomly
  - simulated annealing…
- EM for BNs is a special case of a more general EM algorithm
  - We skip the general case but consider another case: HMMs

# EM for HMMs

- Notation: λ = (A,B, π ), A is the transition model, B is the observation model, π is distribution over the initial state
- Goal is to compute $\lambda^* = \text{argmax}_{\lambda}\ p(O|\ \lambda\ )$
- In the E-step, we will need to compute the following:

$$\gamma_i(t) = p(Q_t = i|O, \lambda)$$

$$\xi_{ij}(t) = p(Q_t = i, Q_{t+1} = j|O, \lambda)$$

- Both can be computed from clique-tree calibration or the specialized forward-backward procedure for HMMs (see next slide)
- $\sum_{t=1}^{T} \gamma_i(t)$ is the expected number of times system is in state *i*, hence also the expected number of transitions away from *i*.
- $\sum_{t=1}^{T-1} \xi_{ij}(t)$ is the expected number of transitions from i to j

# HMM Inferences (from Bilmes Tutorial)

- Define $\alpha_i(t) = p(O_1 = o_1, \ldots, O_t = o_t, Q_t = i|\lambda)$

  1. $\alpha_i(1) = \pi_i b_i(o_1)$
  2. $\alpha_j(t+1) = \left[\sum_{i=1}^{N} \alpha_i(t) a_{ij}\right] b_j(o_{t+1})$
  3. $p(O|\lambda) = \sum_{i=1}^{N} \alpha_i(T)$

- Define $\beta_i(t) = p(O_{t+1} = o_{t+1}, \ldots, O_T = o_T|Q_t = i, \lambda)$

  1. $\beta_i(T) = 1$
  2. $\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_j(t+1)$
  3. $p(O|\lambda) = \sum_{i=1}^{N} \beta_i(1) \pi_i b_i(o_1)$

- Define $\gamma_i(t) = p(Q_t = i|O, \lambda) = \dfrac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}$

$$\xi_{ij}(t) = p(Q_t = i, Q_{t+1} = j|O, \lambda)$$

$$= \frac{\gamma_i(t) a_{ij} b_j(o_{t+1}) \beta_j(t+1)}{\beta_i(t)}$$

# M-step: Recompute Parameters

- Expected frequency in state *i* at time 1

$$\tilde{\pi}_i = \gamma_i(1)$$

- Expected number of transitions from i to j compared to total number of transitions away from i

$$\tilde{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

- Expected number of times observation has been $v_k$, in state *i*, compared to the total number of times in state *i*.

$$\tilde{b}_i(k) = \frac{\sum_{t=1}^{T} \delta_{o_t, v_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

# Next Class

- Read sections  20.2 and 20.3 of the KF book