Probabilistic Graphical Models

Lecture 17 – EM

CS/CNS/EE 155
Andreas Krause

Announcements

- Project poster session on **Thursday Dec 3, 4-6pm in Annenberg 2nd floor atrium!**
  - Easels, poster boards and cookies will be provided!

- Final writeup (8 pages NIPS format) due Dec 9

Approximate inference

- Three major classes of general-purpose approaches

- **Message passing**
  - E.g.: Loopy Belief Propagation (today!)

- **Inference as optimization**
  - Approximate posterior distribution by simple distribution
  - Mean field / structured mean field
  - Assumed density filtering / expectation propagation

- **Sampling based inference**
  - Importance sampling, particle filtering
  - Gibbs sampling, MCMC

- Many other alternatives (often for special cases)

Sample approximations of expectations

- $x_1,\ldots,x_N$ samples from RV X
- Law of large numbers:

$$\mathbb{E}_P[f(X)] = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

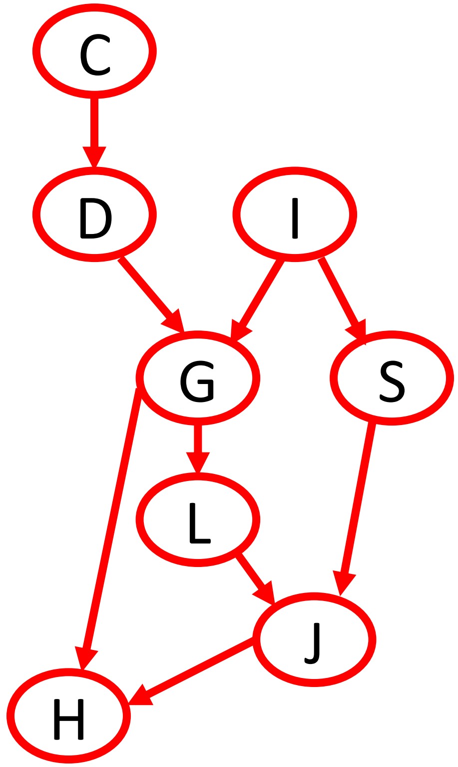
- Hereby, the convergence is with probability 1 (almost sure convergence)
- Finite samples:

$$\mathbb{E}_P[f(X)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Monte Carlo sampling from a BN

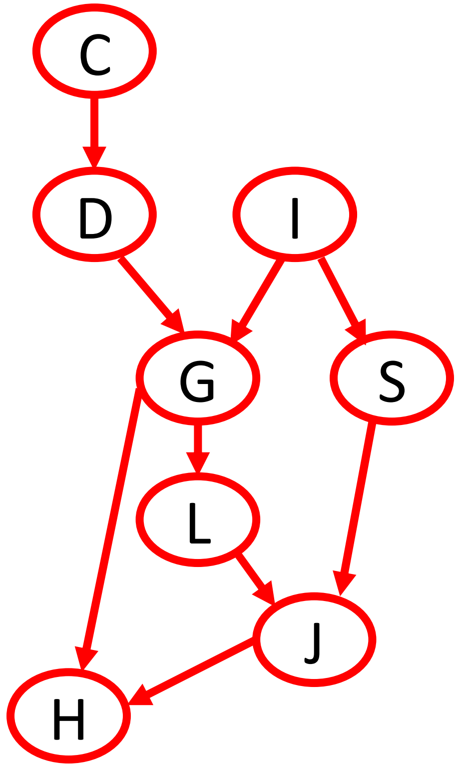
- Sort variables in topological ordering $X_1,\ldots,X_n$
- For i = 1 to n do
  - Sample $x_i \sim P(X_i \mid X_1=x_1, \ldots, X_{i-1}=x_{i-1})$ $= P(X_i \mid Pa_{X_i})$

- Works even with high-treewidth models!



Computing probabilities through sampling

- Want to estimate probabilities
- Draw N samples from BN



- Marginals

$$P(H=y) = \mathbb{E}_P[I_{H=y}] = \sum_x P(x) \cdot \underbrace{I_{H=y}(x)}_{=1 \text{ iff } x_H = y}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} I_{H=y}(x^{(i)}) = \frac{Count(H=y)}{N}$$

- Conditionals

$$P(D=h \mid H=m) = \frac{P(D=h, H=m)}{P(H=m)} = \frac{Count(D=h, H=m)}{Count(H=m)}$$

Rejection sampling

- Rejection sampling problematic for rare events

Sampling from intractable distributions

$P(X_A \mid X_B = x_B) \propto P(X_A, X_B = x_B)$

- Given unnormalized distribution

  $P(X) \propto \underline{Q(X)} = P(X, X_{obs} = x_{obs})$

- Q(X) efficient to evaluate, but normalizer intractable
- For example, $Q(X) = \underline{\prod_j \Psi(C_j)}$

- Want to sample from P(X) $= \frac{1}{Z} Q(x)$

- **Ingenious idea**:
  Can create Markov chain that is efficient to simulate and that has stationary distribution P(X)

Markov Chain Monte Carlo

- Given an unnormalized distribution Q(x)
- Want to design a Markov chain with stationary distribution

$$\pi(x) = 1/Z\ Q(x)$$

- Need to specify transition probabilities P(x | x')!

Designing Markov Chains

1) Proposal distribution R(X' | X)

- Given $X_t = x$, sample "proposal" $x' \sim R(X' \mid X=x)$
- Performance of algorithm will strongly depend on R

2) Acceptance distribution:

- Suppose $X_t = x$
- With probability $\alpha = \min\left\{1, \frac{Q(x')R(x \mid x')}{Q(x)R(x' \mid x)}\right\}$ set $X_{t+1} = x'$
- With probability $1-\alpha$, set $X_{t+1} = x$

Theorem [Metropolis, Hastings]: The stationary distribution is $Z^{-1}\,Q(x)$

- Proof: Markov chain satisfies detailed balance condition!

Gibbs sampling

- Start with initial assignment $\mathbf{x}^{(0)}$ to all variables
- For t = 1 to ∞ do
  - Set $\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)}$
  - For each variable X_i
    - Set $\mathbf{v}_i$ = values of all $\mathbf{x}^{(t)}$ except x_i
    - Sample $x^{(t)}_i$ from $P(X_i \mid \mathbf{v}_i)$
- Gibbs sampling satisfies detailed balance equation for P
- Can efficiently compute conditional distributions $P(X_i \mid \mathbf{v}_i)$ for graphical models

Summary of Sampling

- Randomized approximate inference for computing expections, (conditional) probabilities, etc.
- Exact in the limit
  - But may need ridiculously many samples
- Can even directly sample from intractable distributions
  - Disguise distribution as stationary distribution of Markov Chain
  - Famous example: Gibbs sampling

Summary of approximate inference

- Deterministic and randomized approaches
- Deterministic
  - Loopy BP
  - Mean field inference
  - Assumed density filtering
- Randomized
  - Forward sampling
  - Markov Chain Monte Carlo
  - Gibbs Sampling

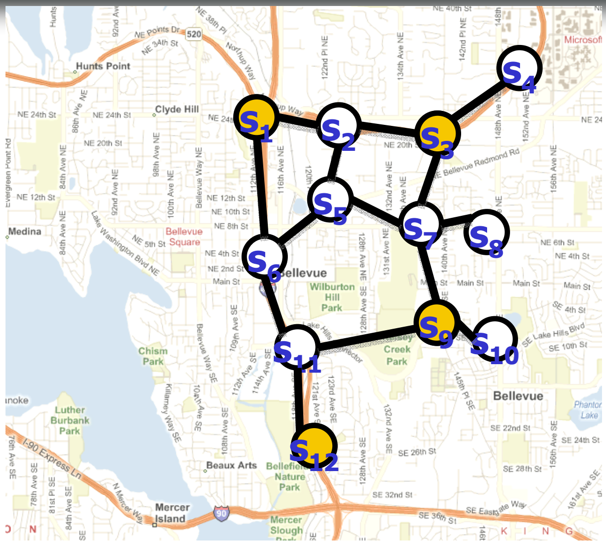
Recall: The “light” side

- Assumed
  - everything fully observable
  - low treewidth
  - no hidden variables

- Then everything is nice ☺
  - Efficient exact inference in large models
  - Optimal parameter estimation without local minima
  - Can even solve some structure learning tasks exactly

The “dark” side

represent



States of the world,
sensor measurements, …

Graphical model

- In the real world, these assumptions are often violated..
- Still want to use graphical models to solve interesting problems..

Remaining Challenges

- Inference
  - **Approximate inference** for high-treewidth models
- Learning
  - Dealing with **missing data**
- Representation
  - Dealing with **hidden variables**

Learning general BNs

	Known structure	Unknown structure
Fully observable	Easy!	Hard
Missing data	Today	

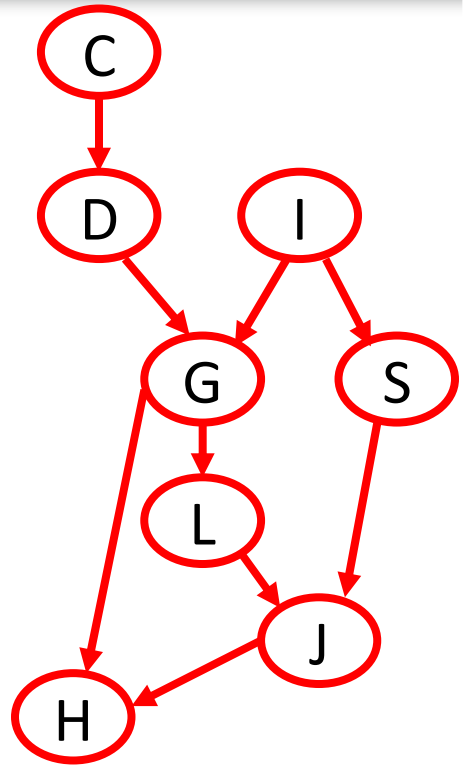
Dealing with missing data

- So far, have assumed all variables are observed in each training example

$x^{(i)} = [C=y, D=h, G=h, I=h, S=l, \ldots]$
$x^{(i+1)} =$

- In practice, often have missing data
  - Some variables may never be observed
  - Missing variables may be different for each example

$x^{(i)} = [G=h, S=l, I=?, L=?, \ldots]$
$x^{(j)} =$



Gaussian Mixture Modeling

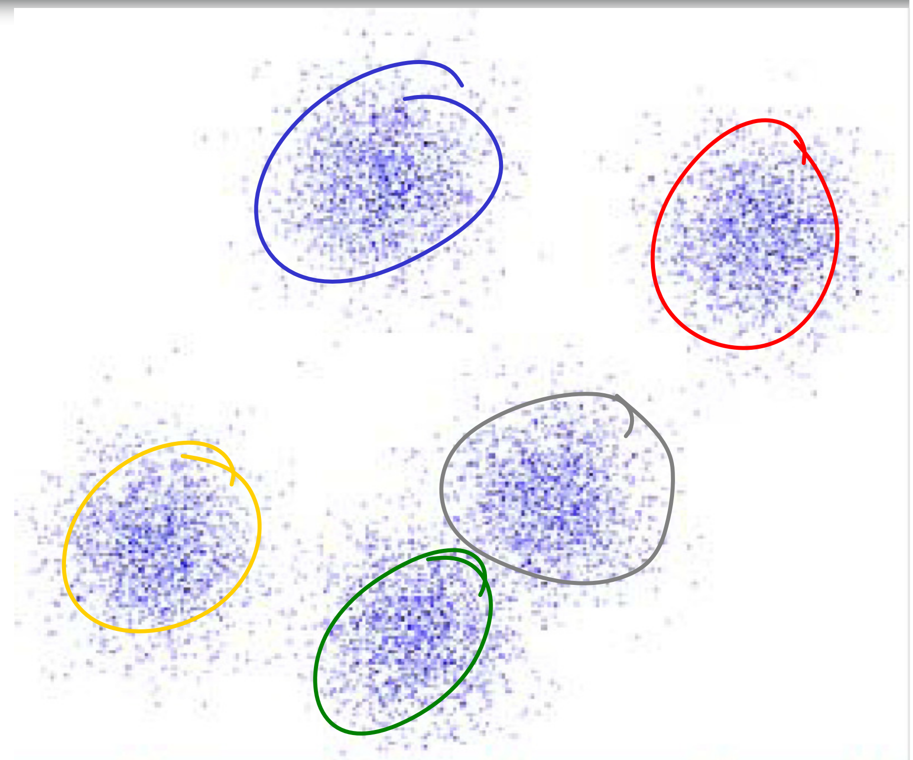
$X = [Y, Z]$

$x^{(1)} = [0.1, .15, \text{blue}]$

$x^{(2)} = [.2, .2, \text{blue}]$

$x^{(3)} = [.7, .7, \text{green}]$

$Z \to Y$



$Z \in \{1, \dots, k\}$

$P(Y=y \mid Z=z) = \mathcal{N}(y; \mu_z, \Sigma_z)$

Learning with missing data

- Suppose **X** is observed variables, **Z** hidden variables
- Training data: $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}$
- Marginal likelihood:

$$P = \prod \psi$$

$$\log P = \sum \log \psi$$

$$\ell(D_x ; \theta) = \sum_{j=1}^{m} \log P(x^{(j)} ; \theta)$$

$$= \sum_{j=1}^{m} \log \sum_{z} P(x^{(j)}, z ; \theta)$$

- Marginal likelihood doesn't decompose

Intuition: EM Algorithm

- Iterative algorithm for parameter learning in case of missing data
- EM Algorithm
  - **E**xpectation Step: “Hallucinate” hidden values
  - **M**aximization Step: Train model as if data were fully observed
  - Repeat
- Will converge to local maximum

E-Step:

- $\mathbf{x}$: observed data; $\mathbf{z}$: hidden data
- "Hallucinate" missing values by computing distribution over hidden variables using current parameter estimate:
- For each example $\mathbf{x}^{(j)}$, compute:

$$Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}^{(j)}) = P(\mathbf{z} \mid \mathbf{x}^{(j)}, \underline{\theta^{(t)}})$$

Current parameter estimate

Towards M-step: Jensen inequality

- Marginal likelihood doesn't decompose

$$\ell(\mathbf{x};\theta) = \sum_j \log \sum_{\mathbf{z}} P(\mathbf{x}^{(j)}, \mathbf{z}; \theta)$$

- **Theorem [Jensen's inequality]**:
  For any distribution P(**z**) and function f(**z**),

$$\log \sum_{\mathbf{z}} P(\mathbf{z}) f(\mathbf{z}) \geq \sum_{\mathbf{z}} P(\mathbf{z}) \log f(\mathbf{z})$$

$$\log\left(\mathbb{E}_P[f(z)]\right) \geq \mathbb{E}_P[\log f(z)]$$

Lower-bounding marginal likelihood

- Jensen's inequality: $\log \sum_{\mathbf{z}} P'(\mathbf{z}) f(\mathbf{z}) \geq \sum_{\mathbf{z}} P'(\mathbf{z}) \log f(\mathbf{z})$
- From E-step: $Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}^{(j)}) = P(\mathbf{z} \mid \mathbf{x}^{(j)}, \theta^{(t)})$

$$\ell(\mathbf{x};\theta) = \sum_j \log \sum_{\mathbf{z}} P(\mathbf{x}^{(j)}, \mathbf{z}; \theta)$$

$$= \sum_j \log \sum_z \underbrace{Q^{(t+1)}(z \mid x^{(j)})}_{P'(z)} \underbrace{\frac{P(x^{(j)}, z; \theta)}{Q^{(t+1)}(z \mid x^{(j)}; \theta)}}_{f(z)}$$

$$\geq \sum_j \sum_z Q^{(t+1)}(z \mid x^{(j)}) \log \frac{P(x^{(j)}, z; \theta)}{Q^{(t+1)}(z \mid x^{(j)}; \theta)}$$

$$= \sum_j \sum_z Q^{(t+1)}(z \mid x^{(j)}) \log P(x^{(j)}, z; \theta) + H(Q^{(t+1)}) \cdot m$$

Lower bound on marginal likelihood

- Bound of marginal likelihood with hidden variables

$$\ell(\mathbf{x};\theta) \geq \sum_{j=1}^{m} \sum_{\mathbf{z}} \underbrace{Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}^{(j)}) \log P(\mathbf{z}, \mathbf{x}^{(j)} \mid \theta)} + \underbrace{mH(Q^{(t+1)})}_{\text{constant}}$$

- Recall: Likelihood in fully observable case:

$$\ell(\mathbf{x};\theta) \geq \sum_{j=1}^{m} \underbrace{\log P(\mathbf{x}^{(j)} \mid \theta)}$$

- Lower-bound interpreted as "weighted" data set

X	Z	Q(z|x)	fully obs:
[0.1, .2]	1	.9	1
[0.1, .2]	2	.1	0
[.5, .4]	1	.3	0
[.5, .4]	2	.7	1

M-step: Maximize lower bound

- Lower bound:

$$\ell(\mathbf{x};\theta) \geq \sum_{j=1}^{m}\sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}^{(j)}) \log P(\mathbf{z}, \mathbf{x}^{(j)} \mid \theta) + mH(Q^{(t+1)})$$

- Choose $\theta^{(t+1)}$ to maximize lower bound

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \sum_{j=1}^{m}\sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}^{(j)}) \log P(\mathbf{z}, \mathbf{x}^{(j)} \mid \theta)$$

- Use expected sufficient statistics (counts). Will see:
  - Whenever we used Count(x,z) in fully observable case, replace by $E_{Q^{t+1}}$[Count(**x**,**z**)]

Coordinate Ascent Interpretation

- Define energy function

$$F[Q,\theta] = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}^{(j)}) \log P(\mathbf{z}, \mathbf{x}^{(j)} \mid \theta) + mH(Q)$$

- For any distribution Q and parameters θ:

$$\ell(\mathbf{x};\theta) \geq F[Q,\theta]$$

- EM algorithm performs coordinate ascent on F:

$$Q^{(t+1)} = \operatorname*{argmax}_{Q} F[Q, \theta^{(t)}]$$

$$\theta^{(t+1)} = \operatorname*{argmax}_{\theta} F[Q^{(t+1)}, \theta]$$

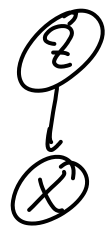
- Monotonically converges to local maximum

EM for Gaussian Mixtures

E-Step

$$Q^{(t+1)}(z \mid x^{(j)})$$

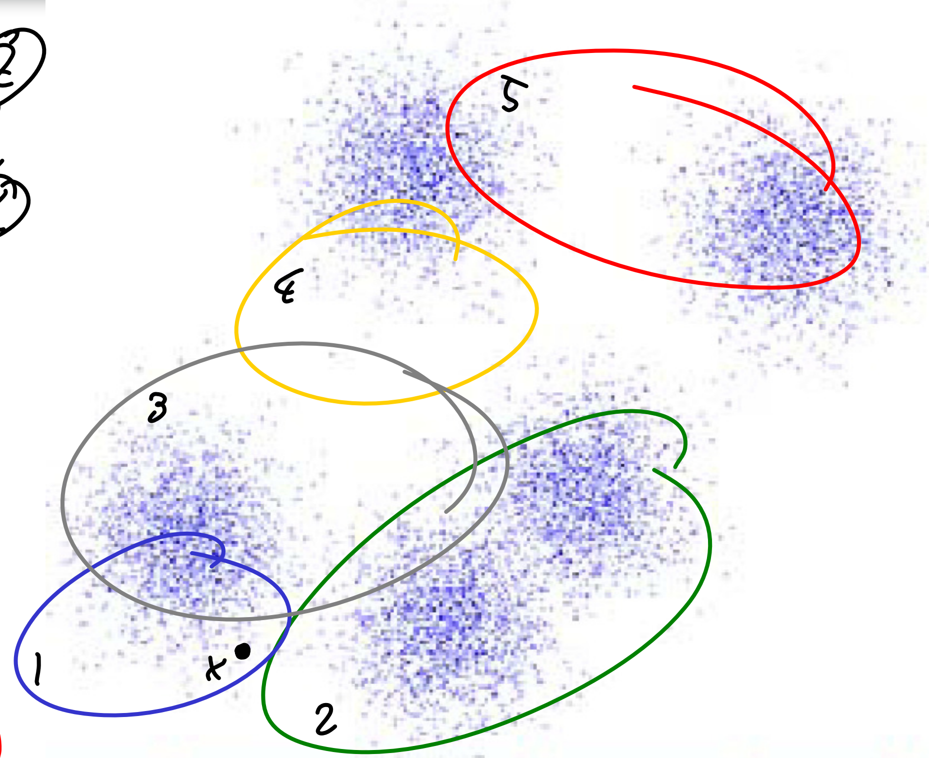
$$= P(z = z \mid x^{(j)}; \theta^{(t)})$$



$$Q^{(1)}(z = 1 \mid x = [.1, .2]) = .4$$

2 .3

3 .3



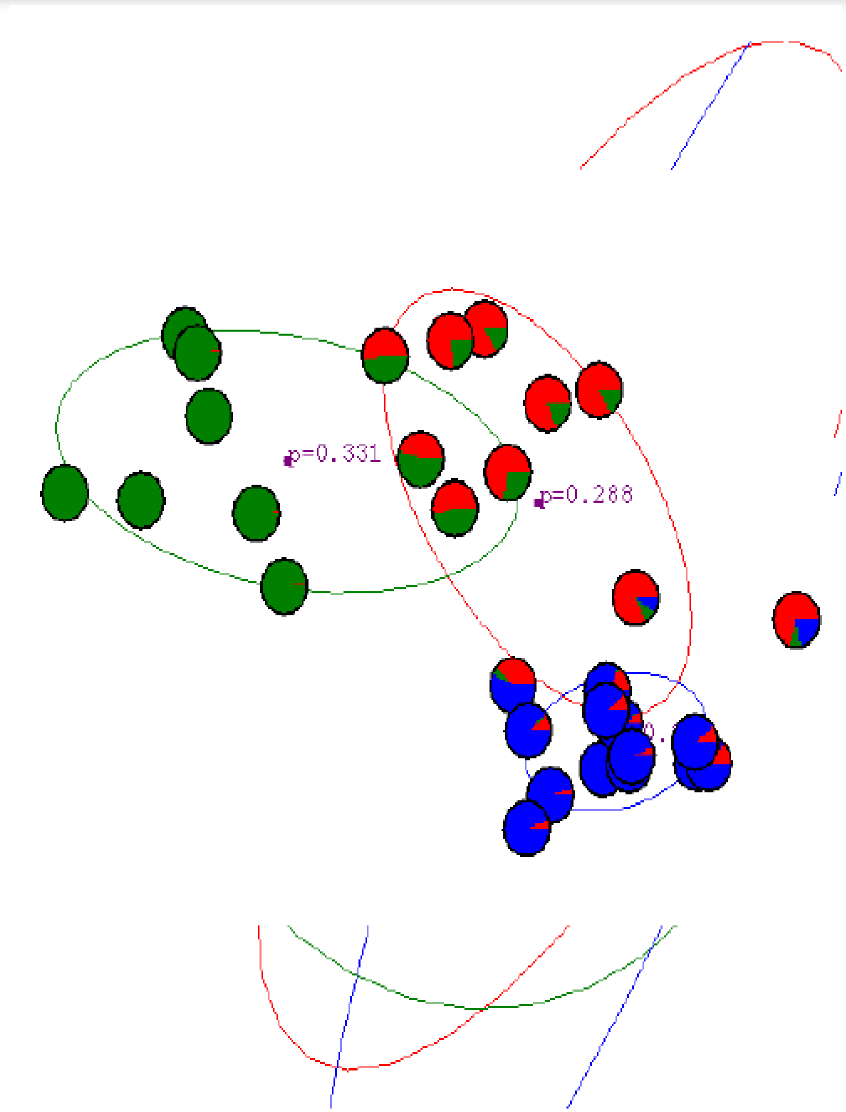
$$P(z \mid x) \propto P(x \mid z) P(z)$$

M-Step:

$$\mu_i^{(t+1)} = \frac{\sum_{j=1}^{m} Q(z=i \mid x^{(j)}) \cdot x^{(j)}}{\sum_{j=1}^{m} Q(z=i \mid x^{(j)})}$$

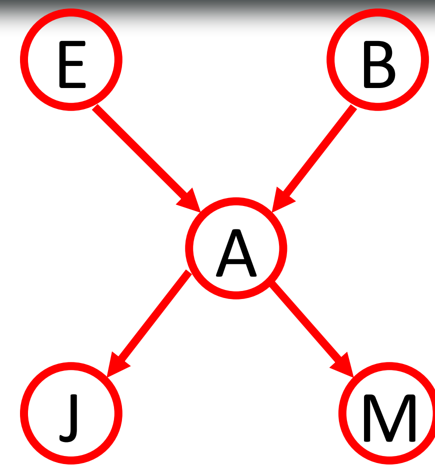
$$\Sigma_i^{(t+1)} = \ldots$$

EM Iterations [by Andrew Moore]



EM in Bayes Nets

- Complete data likelihood



$$\ell(D;\theta) = \sum_j \log P(e^{(j)}|\theta) \cdot P(b^{(j)}|\theta) \cdot P(a^{(j)}|\theta), \ldots$$

$$= \sum_j \log \prod_i P(X_i | Pa_i)$$

$$= \sum_j \sum_i \log P(X_i | Pa_i)$$

Decomposes

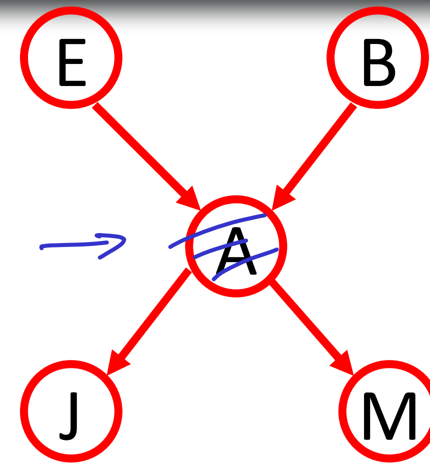
Can optimize each CPT independently!

EM in Bayes Nets

- Incomplete data likelihood

$$\ell(D;\theta) = \sum_j \log \sum_a P(e^{(j)}|\theta)\, P(b^{(j)}|\theta)\, P(a^{(j)}|e,b) \cdots$$

unobs $\longrightarrow$

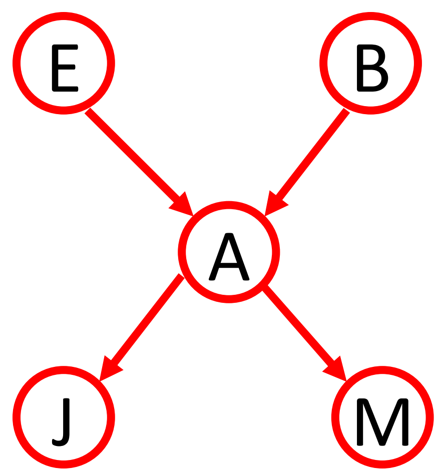


Does not decompose :(

E-Step for BNs

- Need to compute $Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}^{(j)}) = P(\mathbf{z} \mid \mathbf{x}^{(j)}, \theta^{(t)})$
- For fixed **z**, **x**: Can compute using inference
- Naively specifying full distribution would be intractable

Have to use $Q^{(t+1)}$ "implicitly" (as needed)



M-step for BNs

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}^{(j)}) \log P(\mathbf{z}, \mathbf{x}^{(j)} \mid \theta)$$

- Can optimize each CPT independently!

- MLE in fully observed case:

$$\widehat{\theta}_{x|\mathbf{pa}_x} = \frac{\mathrm{Count}(x, \mathbf{pa}_x)}{\mathrm{Count}(\mathbf{pa}_x)}$$

- MLE with hidden data:

$$\widehat{\theta}_{x|\mathbf{pa}_x}^{(t+1)} = \frac{\mathbb{E}_{Q^{(t+1)}}[\mathrm{Count}(x, \mathbf{pa}_x)]}{\mathbb{E}_{Q^{(t+1)}}[\mathrm{Count}(\mathbf{pa}_x)]}$$

Computing expected counts

$$\widehat{\theta}^{(t+1)}_{x|\mathbf{pa}_x} = \frac{\mathbb{E}_{Q^{(t+1)}}[\text{Count}(x, \mathbf{pa}_x)]}{\mathbb{E}_{Q^{(t+1)}}[\text{Count}(\mathbf{pa}_x)]}$$

- Suppose we observe O=o
- Variables A hidden

Partition A into A_o, A_h : $A_o \subset O$; $A_h \cap O = \emptyset$

$$\mathbb{E}_Q[\text{Count}(A_o = a_o', A_h = a_h')]$$

$$= \sum_j \mathbb{I}[a_o^{(j)} = a_o'] \cdot \underbrace{Q(a_h \mid O = o^{(j)})}$$

To evaluate need to perform inference

1 Inference per data point

Learning general BNs

	Known structure	Unknown structure
Fully observable	Easy!	Hard (2.)
Missing data	**EM**	**Now**

Structure learning with hidden data

- Fully observable case:
  - Score(D;G) = likelihood of data under most likely parameters
  - Decomposes over families
    Score(D;G) = $\sum_i$ FamScore$_i$(X_i | Pa_{X_i})
  - Can recompute score efficiently after adding/removing edges

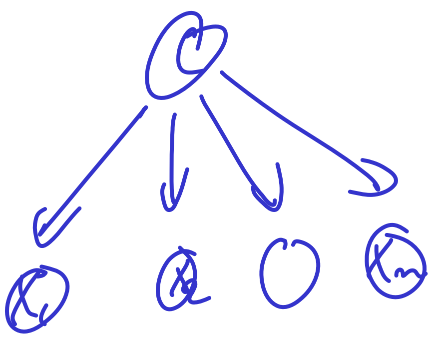
- Incomplete data case:
  - Score(D;G) = lower bound from EM
  - Does not decompose over families
  - Search is very expensive

- Structure-EM: Iterate
  - Computing of expected counts
  - Multiple iterations of structure search for fixed counts
- Guaranteed to monotonically improve likelihood score

Hidden variable discovery

- Sometimes, "invention" of a hidden variable can drastically simplify model

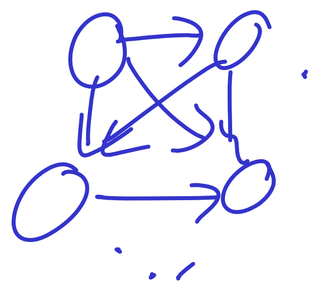
"True" world



We only know about/model

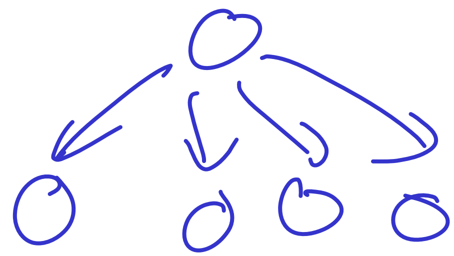
$X_1, \ldots, X_m$

Best fit to data:

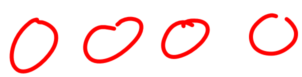


"Guess" existence of hidden variable & and run structure EM

⇒ (hopefully)* recover



But: Can't identify common effects



⇒

⇒ Strong limits to identifiability

Learning general BNs

	Known structure	Unknown structure
Fully observable	Easy!	Hard (2.)
Missing data	**EM**	**Structure-EM**