ECE 6504: Advanced Topics in Machine Learning

Probabilistic Graphical Models and Large-Scale Learning

Topics
- BN / MRFs
  - Learning from hidden data
  - EM

Readings: KF 19.1-3, Barber 11.1-2

Dhruv Batra

Virginia Tech

Administrativia

- (Mini-)HW4
  - Out now
  - Due: May 7, 11:55pm
  - Implementation:
    - Parameter Learning with Structured SVMs and Cutting-Plane

- Final Project Webpage
  - Due: May 7, 11:55pm
  - Can use late days
  - 1-3 paragraphs
    - Goal
    - Illustrative figure
    - Approach
    - Results (with figures or tables)

- Take Home Final
  - Out: May 8
  - Due: May 13, 11:55pm
  - No late days
  - Open book, open notes, open internet. Cite your sources.
  - No discussions!

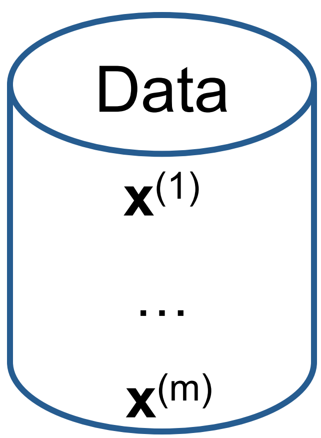
Recap of Last Time

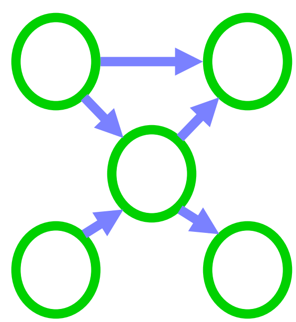
Main Issues in PGMs

- Representation
  - How do we store $P(X_1, X_2, \ldots, X_n)$
  - What does my model mean/imply/assume? (Semantics)
- Inference
  - How do I answer questions/queries with my model? such as
  - Marginal Estimation: $P(X_5 \mid X_1, X_4)$
  - Most Probable Explanation: argmax $P(X_1, X_2, \ldots, X_n)$
- Learning
  - How do we learn parameters and structure of $P(X_1, X_2, \ldots, X_n)$ from data?
  - What model is the right for my data?

Learning Bayes Nets

	Known structure	Unknown structure
Fully observable data	Very easy	Hard
Missing data	Somewhat easy (EM)	Very very hard





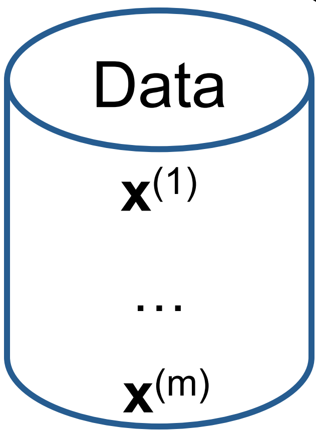
structure

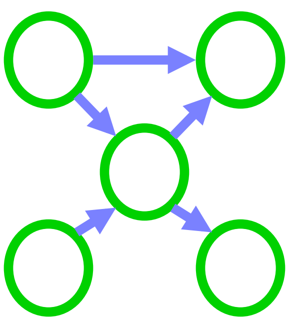
+

CPTs – $P(X_i | \mathbf{Pa}_{X_i})$

parameters

Learning the CPTs



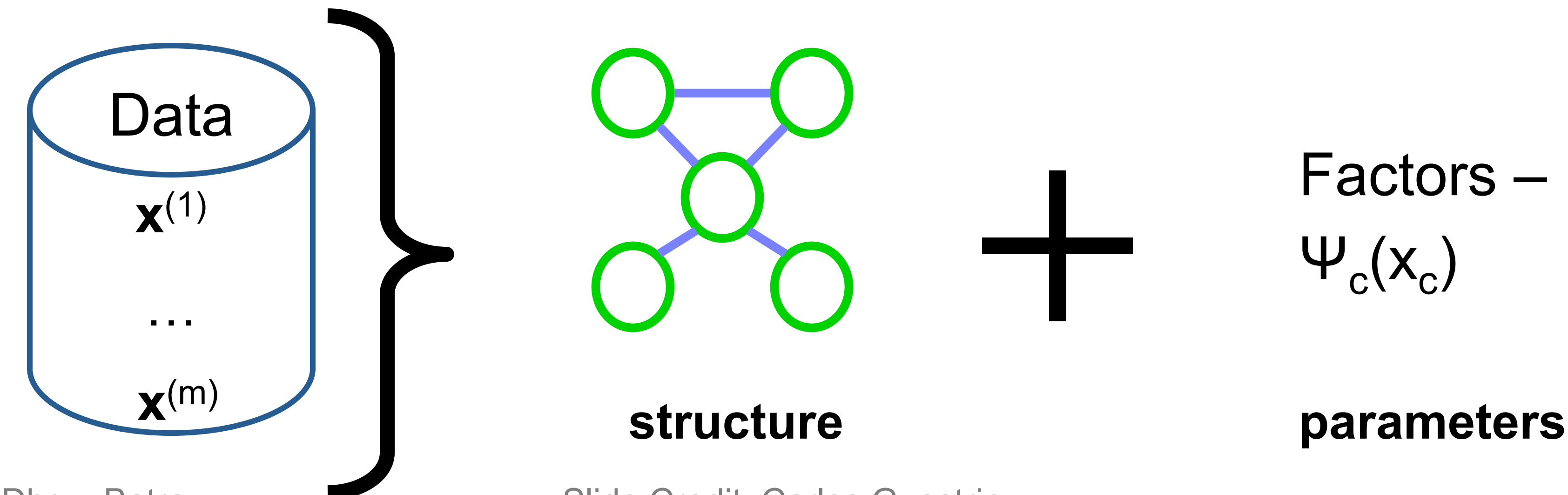


For each discrete variable X_i

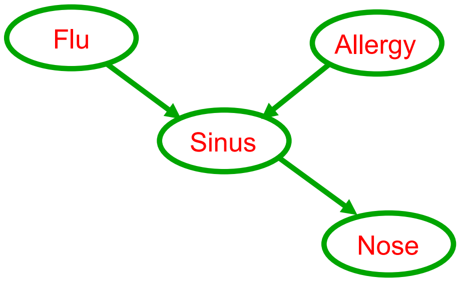
$$\hat{P}_{MLE}(X_i = a \mid \mathrm{Pa}_{X_i} = b) = \frac{\mathrm{Count}(X_i = a, \mathrm{Pa}_{X_i} = b)}{\mathrm{Count}(\mathrm{Pa}_{X_i} = b)}$$

Learning Markov Nets

	Known structure	Unknown structure
Fully observable data	NP-Hard (but doable)	Harder
Missing data	Harder (EM)	Don't try this at home



Learning Parameters of a BN



- Log likelihood decomposes:

$$\log P(\mathcal{D} \mid \theta) = m \sum_i \sum_{x_i, \mathbf{Pa}_{x_i}} \widehat{P}(x_i, \mathbf{Pa}_{x_i}) \log P(x_i \mid \mathbf{Pa}_{x_i})$$

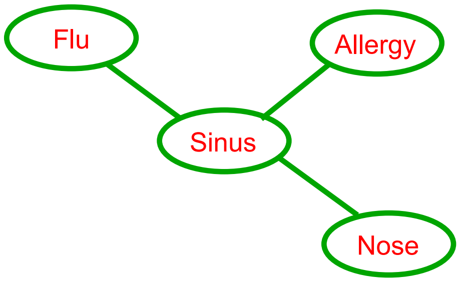
- Learn each CPT independently

$$\widehat{P}(\mathbf{u}) = \frac{\text{Count}(\mathbf{U} = \mathbf{u})}{m}$$

- Use counts

Log Likelihood for MN

- Log likelihood decomposes:



$$\log P(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \sum_{\mathbf{c}_i} \widehat{P}(\mathbf{c}_i) \log \psi_i(\mathbf{c}_i) - m \log Z$$

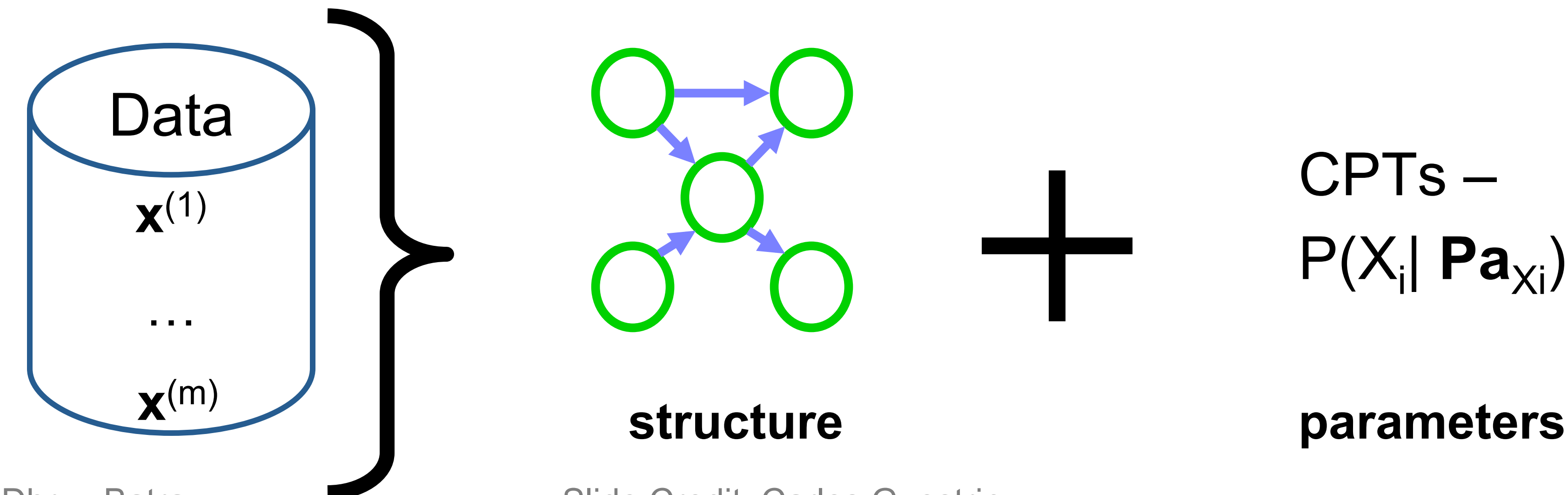
- Doesn't decompose!
  – logZ couples all parameters together

Plan for today

- BN Parameter Learning with Missing Data
  - Why model latent variables?
  - Expectation Maximization (EM)

Learning Bayes Nets

	Known structure	Unknown structure
Fully observable data	Very easy	Hard
Missing data	Somewhat easy (EM)	Very very hard



When is data missing?

- Fully Observed Data
- Some hidden variables
  - Never observed
- General hidden pattern
  - Arbitrary entries missing in the data matrix

Why missing data?

- Sometimes no choice
  - sensor error, some data dropped
  - Data collection error, we forgot to ask this question

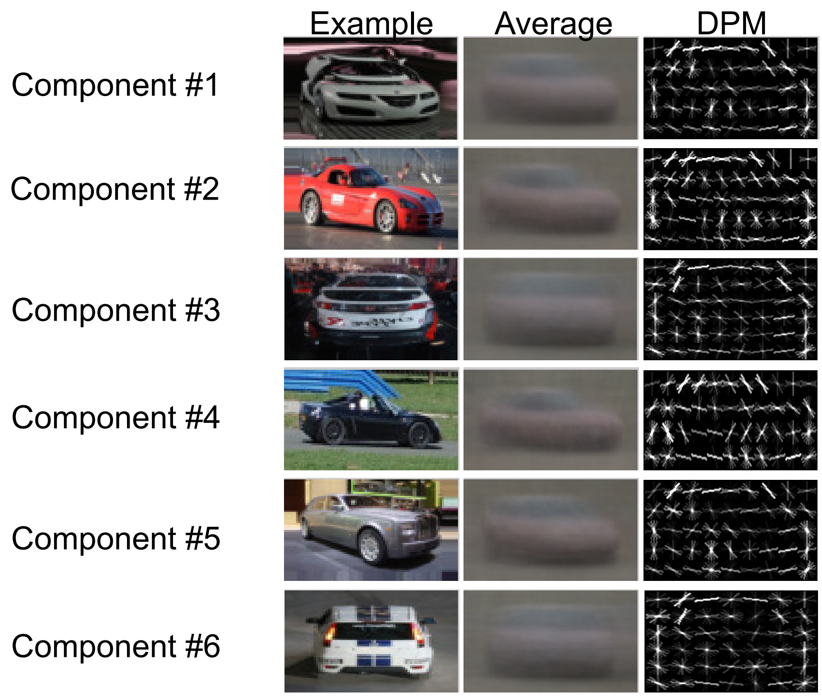
Why *introduce* hidden variables?

- Model Sparsity!
  - Modeling hidden/latent variables can simplify interactions
  - Reduction in #parameters to be learnt

- Example
  - On board

Why *introduce* hidden variables?

- Discovering Clusters in data!
  - Modeling different P(y|x,h) for each h



Treating Missing Data

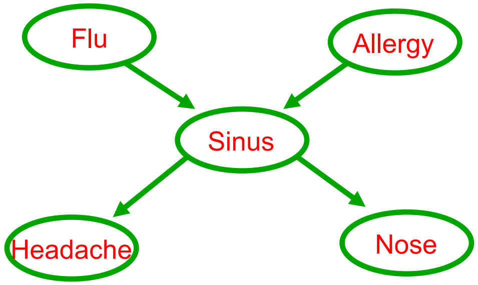
- Thought Experiment:
  - Coin Toss: H,T,?,?,H,H,?
- Case 1: Missing at Random
- Case 2: Missing with bias
- BN illustration of the two cases
  - On board
  - Takeaway message: Need to *model* missing data

Likelihood with Complete/Missing Data

- Example on board X->Y
  - One variable X; parameter θ_X
  - Two variables X,Y; parameters θ_X, $\theta_{Y|X}$

- Takeaway Messages:
  - Parameters get coupled (LL = sum-log-sum doesn't factorize)
  - Computing LL requires marginal inference!

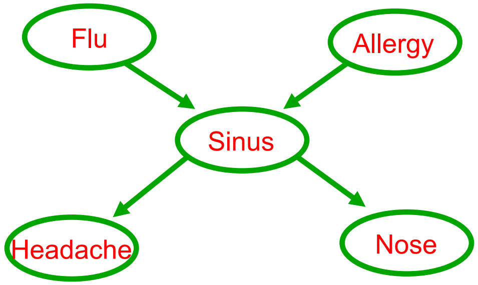
Data likelihood for BNs



- Given structure, log likelihood of fully observed data:

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G})$$

Marginal likelihood

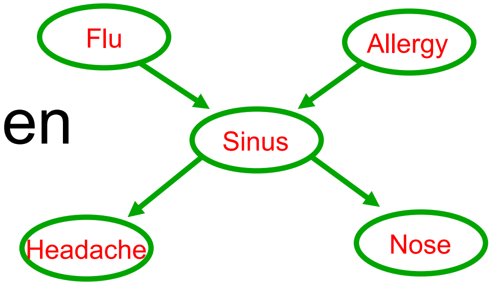


- What if S is hidden?

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G})$$

Log likelihood for BNs with hidden data

- Marginal likelihood – **O** is observed, **H** is hidden



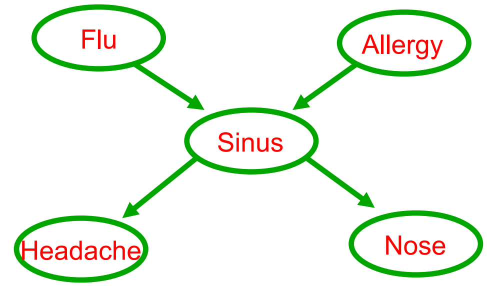
$$\ell(\theta : \mathcal{D}) = \sum_{j=1}^{m} \log P(\mathbf{o}^{(j)} \mid \theta)$$
$$= \sum_{j=1}^{m} \log \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{o}^{(j)} \mid \theta)$$

EM Intuition

- Chicken & Egg problem
  - If we knew h, then learning θ would be easy
  - If we knew theta, then finding P(h | o, θ) would be "easy"
    - Sum-product inference

- EM solution
  - Initialize
  - Fix θ, find P(h | o, θ)
  - Use these to learn θ

E-step for BNs

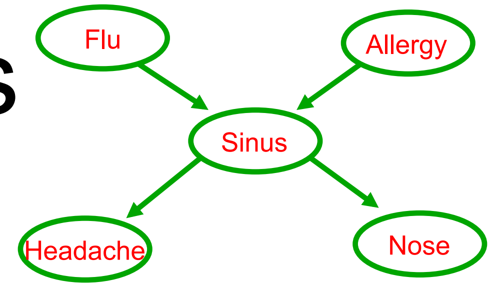


- E-step computes probability of hidden vars **h** given **o**

$$Q^{(t+1)}(\mathbf{h} \mid \mathbf{o}) \leftarrow P(\mathbf{h} \mid \mathbf{o}, \theta^{(t)})$$

- Corresponds to inference in BN

The M-step for BNs

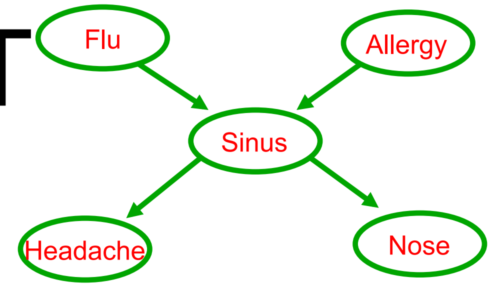


- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{j=1}^{m} \sum_{\mathbf{h}} Q^{(t+1)}(\mathbf{h} \mid \mathbf{o}^{(j)}) \log P(\mathbf{h}, \mathbf{o}^{(j)} \mid \theta)$$

- Use expected counts instead of counts:
  - If learning requires Count(**h**,**o**)
  - Use $E_{Q(t+1)}$[Count(**h**,**o**)]

M-step for each CPT



- M-step decomposes per CPT
  - Standard MLE:

$$P(X_i = x_i \mid \mathbf{Pa}_{X_i} = \mathbf{z}) = \frac{\text{Count}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z})}{\text{Count}(\mathbf{Pa}_{X_i} = \mathbf{z})}$$

$$P(X_i = x_i \mid \mathbf{Pa}_{X_i} = \mathbf{z}) = \frac{\text{ExCount}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z})}{\text{ExCount}(\mathbf{Pa}_{X_i} = \mathbf{z})}$$

The general learning problem with missing data

- Marginal likelihood – **o** is observed, **h** is missing:

$$
\begin{aligned}
ll(\theta : \mathcal{D}) &= \log \prod_{j=1}^{M} P(\mathbf{o}^j \mid \theta) \\
&= \sum_{j=1}^{M} \log P(\mathbf{o}^j \mid \theta) \\
&= \sum_{j=1}^{M} \log \sum_{\mathbf{h}} P(\mathbf{o}^j, \mathbf{h} \mid \theta)
\end{aligned}
$$

Applying Jensen's inequality

- Use: $\log \sum_{\mathbf{h}} P(\mathbf{h})\, f(\mathbf{h}) \geq \sum_{\mathbf{h}} P(\mathbf{h}) \log f(\mathbf{h})$

$$ll(\theta : \mathcal{D}) = \sum_{j=1}^{M} \log \sum_{\mathbf{h}} Q_j(\mathbf{h}) \frac{P(\mathbf{o}^j, \mathbf{h} \mid \theta)}{Q_j(\mathbf{h})}$$

Convergence of EM

- Define potential function F(θ,Q):

$$ll(\theta : \mathcal{D}) \;\geq\; F(\theta, Q_j) \;=\; \sum_{j=1}^{M} \sum_{\mathbf{h}} Q_j(\mathbf{h}) \log \frac{P(\mathbf{o}^j, \mathbf{h} \mid \theta)}{Q_j(\mathbf{h})}$$

- EM corresponds to coordinate ascent on F
  - Fix θ, maximize Q
  - Fix Q, maximize θ
  - Thus, maximizes lower bound on marginal log likelihood

EM is coordinate ascent

$$ll(\theta : \mathcal{D}) \geq F(\theta, Q_j) = \sum_{j=1}^{M} \sum_{\mathbf{h}} Q_j(\mathbf{h}) \log \frac{P(\mathbf{o}^j, \mathbf{h} \mid \theta)}{Q_j(\mathbf{h})}$$

- **E-step**: Fix $\theta^{(t)}$, maximize F over Q:
  - On board
  - "Realigns" F with likelihood: $Q_j(\mathbf{h}) = P(\mathbf{h} \mid \mathbf{o}^j, \theta^{(t)})$

$$F(\theta^{(t)}, Q^{(t)}) = ll(\theta^{(t)} : \mathcal{D})$$

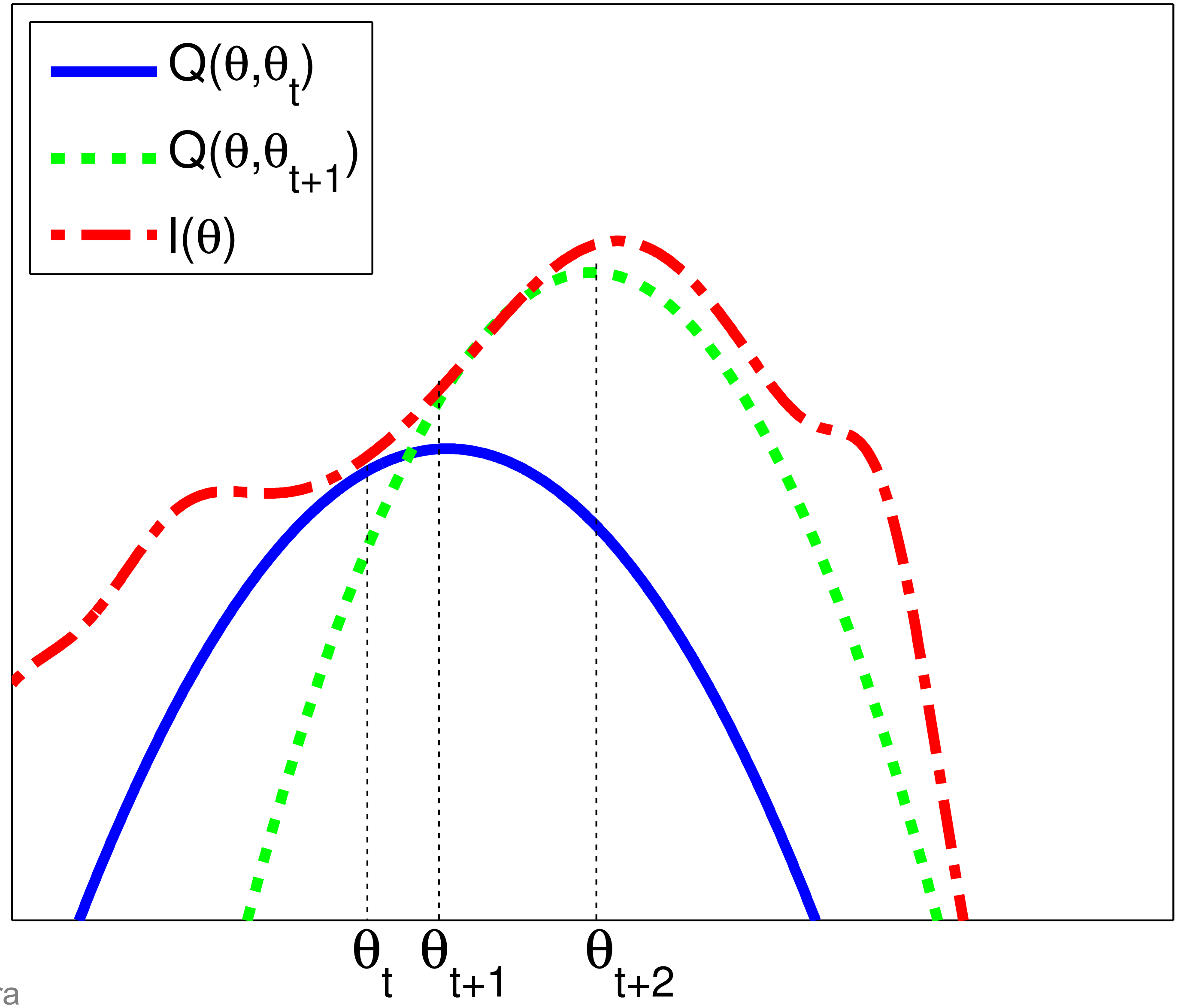
EM is coordinate ascent

$$ll(\theta : \mathcal{D}) \;\geq\; F(\theta, Q_j) \;=\; \sum_{j=1}^{M} \sum_{\mathbf{h}} Q_j(\mathbf{h}) \log \frac{P(\mathbf{o}^j, \mathbf{h} \mid \theta)}{Q_j(\mathbf{h})}$$

- **M-step**: Fix $Q^{(t)}$, maximize F over θ

- Corresponds to weighted dataset:
  - <$\mathbf{o}^1$,$\mathbf{h}$=1> with weight $Q^{(t+1)}(\mathbf{h}=1|\mathbf{o}^1)$
  - <$\mathbf{o}^1$,$\mathbf{h}$=2> with weight $Q^{(t+1)}(\mathbf{h}=2|\mathbf{o}^1)$
  - <$\mathbf{o}^1$,$\mathbf{h}$=3> with weight $Q^{(t+1)}(\mathbf{h}=3|\mathbf{o}^1)$
  - <$\mathbf{o}^2$,$\mathbf{h}$=1> with weight $Q^{(t+1)}(\mathbf{h}=1|\mathbf{o}^2)$
  - <$\mathbf{o}^2$,$\mathbf{h}$=2> with weight $Q^{(t+1)}(\mathbf{h}=2|\mathbf{o}^2)$
  - <$\mathbf{o}^2$,$\mathbf{h}$=3> with weight $Q^{(t+1)}(\mathbf{h}=3|\mathbf{o}^2)$
  - …

EM Intuition



What you need to know about learning BNs with missing data

- EM for Bayes Nets
- E-step: inference computes expected counts
  - Only need expected counts over X_i and $\mathbf{Pa}_{xi}$
- M-step: expected counts used to estimate parameters
- Which variables are hidden can change per datapoint
  - Also, use labeled and unlabeled data → some data points are complete, some include hidden variables