

Readings:

K&F: 11.3, 11.5

Yedidia et al. paper from the class website

## Mean Field and Variational Methods

### Loopy Belief Propagation

Graphical Models – 10708

Carlos Guestrin

Carnegie Mellon University

November 8<sup>th</sup>, 2006

1

## Understanding Reverse KL, Energy Function & The Partition Function

$$\ln Z = \underbrace{F[P_{\mathcal{F}}, Q]}_{\text{constant}} + \underbrace{D(Q||P_{\mathcal{F}})}_{\text{min}} \quad \uparrow F[P_{\mathcal{F}}, Q] = \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi] + H_Q(\mathcal{X})$$

- Maximizing Energy Functional  $\Leftrightarrow$  Minimizing Reverse KL

$$\uparrow \underbrace{F[P_{\mathcal{F}}, Q]}_{\text{max}} \Leftrightarrow \downarrow \underbrace{D(Q||P_{\mathcal{F}})}_{\text{min}}, \forall Q \ D(Q||P_{\mathcal{F}}) \geq 0$$

- **Theorem:** Energy Function is lower bound on partition function

$$\ln Z \geq \underbrace{F[P_{\mathcal{F}}, Q]}_{\text{lower bound}} \uparrow \text{maximize}$$

$$\ln Z = F[P_{\mathcal{F}}, Q] \text{ iff } Q = P_{\mathcal{F}}$$

- Maximizing energy functional corresponds to search for tight lower bound on partition function

# Structured Variational Approximate Inference

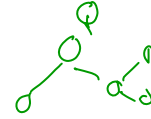
$$\ln Z = F[P_{\mathcal{F}}, Q] + D(Q || P_{\mathcal{F}})$$

$$F[P_{\mathcal{F}}, Q] = \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi] + H_Q(x)$$

- Pick a family of distributions Q that allow for exact inference

e.g., graphical model for Q

- e.g., fully factorized (mean field)  $Q(x) = \prod_i Q_i(x_i)$



- Find  $Q \in \mathcal{Q}$  that maximizes  $F[P_{\mathcal{F}}, Q]$

$F$  is graphical model for P

- For mean field:  $F(P_{\mathcal{F}}, Q) = \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi] + \sum_i H_Q(x_i)$
- max  $Q_i$*
- expected value* (under  $E_Q$ )
- node entropy* (for  $H_Q(x_i)$ )
- $\forall x_i: \sum_{x_i} Q_i(x_i) = 1$      $Q_i(x_i) \geq 0 \forall x_i$

# Optimization for mean field

$$\max_Q F[P_{\mathcal{F}}, Q] = \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi] + \sum_j H_{Q_j}(x_j), \quad \forall i, \sum_{x_j} Q_j(x_j) = 1$$

- Constrained optimization, solved via Lagrangian multiplier

- $\exists \lambda$ , such that optimization equivalent to:

$$L(Q, \lambda) = \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi] + \sum_j H_{Q_j}(x_j) + \lambda \left( \sum_{x_j} Q_j(x_j) - 1 \right)$$

- Take derivative, set to zero

*local minima* (pointing to the  $\lambda$  term)

*maxima* (pointing to the entropy term)

*saddle points* (pointing to the  $\lambda$  term)

*use 1st & 2nd* (pointing to the  $\lambda$  term)

- **Theorem:** Q is a stationary point of mean field approximation iff for each i:

$$Q_i(x_i) = \frac{1}{Z_i} \exp \left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi | x_i] \right\}$$

# Understanding fixed point equation

$$Q_i(x_i) = \frac{1}{Z_i} \exp \left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi | x_i] \right\}$$

$$E_Q[\ln \phi | x_i] = \sum_x Q(x_i) \log \phi(x_i)$$

$$= \sum_{i \in I} Q(I=i | G=g) \log \phi(G=g, I=i)$$

$$= \sum_{i \in I} Q(I=i) \log \phi(G=g, I=i)$$

$$= 0.8 \log 1 + 0.2 \log 2$$

$$\phi(G, I) = \begin{array}{c|cc} & t & f \\ \hline t & 1 & 2 \\ \hline f & 3 & 4 \end{array}$$

$$Q(I) = \begin{array}{c|cc} & t & f \\ \hline & 0.8 & 0.2 \end{array}$$

$$P_F(x) = \frac{1}{Z} \prod_i \phi_i(x_i)$$

$$Q(G, I) = \phi(G, I)$$

$$Q(I=i | G=g) = Q(I=i)$$

# Simplifying fixed point equation

$$Q_i(x_i) = \frac{1}{Z_i} \exp \left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi | x_i] \right\}$$

$$E_Q[\ln \phi | x_i = x_i]$$

Suppose  $\phi_{HS}(H, S) : E_Q[\ln \phi_{HS} | G=g] = \sum_{h,j} Q(h,j|g) \cdot \ln \phi(h,j)$

$$= E_Q[\ln \phi_{HS}] = \text{constant}$$

plays no role in fixed point eq.

$$Q = \prod_i Q_i(x_i)$$

only  $\phi$ 's where  $x_i \in \text{Scope}[\phi]$  play a role in f.p. eqn. for  $x_i$

$$Q(h,j|g) = Q(h,j)$$

# $Q_i$ only needs to consider factors that intersect $X_i$

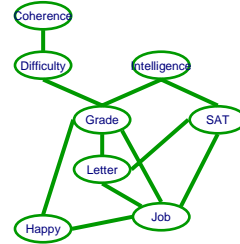
■ **Theorem:** The fixed point:

$$Q_i(x_i) = \frac{1}{Z_i} \exp \left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi | x_i] \right\}$$

is equivalent to:

$$Q_i(x_i) = \frac{1}{Z_i} \exp \left\{ \sum_{\phi_j: X_i \in \text{Scope}[\phi_j]} E_Q[\ln \phi_j(\mathbf{U}_j, x_i)] \right\}$$

□ where the  $\text{Scope}[\phi_j] = \mathbf{U}_j \cup \{X_i\}$



# There are many stationary points!

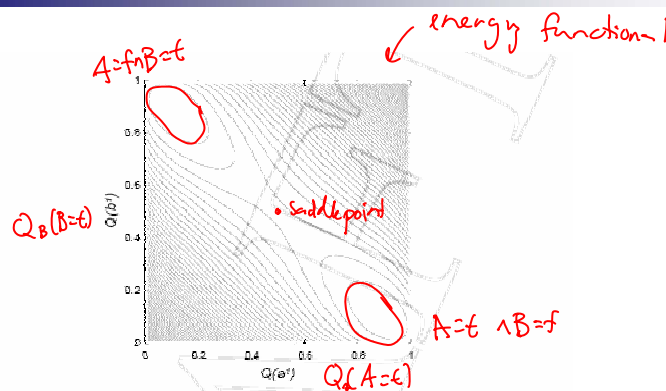
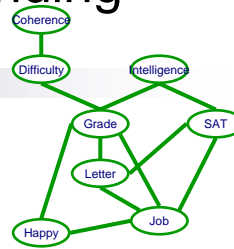


Figure 11.18 An example of a multi-modal mean field energy functional landscape. In this network,  $P(a, b) = 0.25 - \epsilon$  if  $a \neq b$  and  $\epsilon$  if  $a = b$ . The axes correspond to the mean field marginal for  $A$  and  $B$  and the contours show equi-values of the energy functional.

# Very simple approach for finding one stationary point

- Initialize  $Q$  (e.g., randomly or smartly)
- Set all vars to unprocessed
- Pick unprocessed var  $X_i$ 
  - update  $Q_i$ :
 
$$Q_i^{new}(x_i) = \frac{1}{Z_i} \exp \left\{ \sum_{\phi_j: X_i \in \text{Scope}[\phi_j]} E_Q[\ln \phi_j(U_j, x_i)] \right\}$$
  - set var  $i$  as processed
  - if  $Q_i$  changed
    - set neighbors of  $X_i$  to unprocessed
- Guaranteed to converge



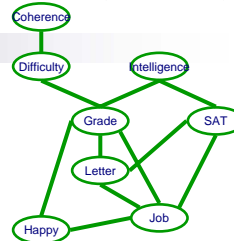
# More general structured approximations

- Mean field very naïve approximation
- Consider more general form for  $Q$ :
 
$$Q(x) = \frac{1}{Z} \prod_i \psi_i(c_i)$$
  - assumption: exact inference doable over  $Q$
- **Theorem:** stationary point of energy functional:

$$\psi_j(c_j) \propto \exp \left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi | c_j] - \sum_{\psi \in \mathcal{Q} \setminus \{\psi_j\}} E_Q[\ln \psi | c_j] \right\}$$

↑ expectation w.r.t.  $Q$ 
↑ exp. w.r.t.  $Q$

want to fit model  $\phi$ 
"removing double counting" comes from  $H_q$



eg:  $Q$ : tree



# Computing update rule for general case

$$\psi_j^{new}(c_j) \propto \exp \left\{ \sum_{\phi \in \mathcal{F}} E_Q^{old}[\ln \phi | c_j] - \sum_{\psi \in \mathcal{Q} \setminus \{\psi_j\}} E_Q^{old}[\ln \psi | c_j] \right\}$$

■ Consider one  $\phi$ :  $\phi(L, S, J)$  ;  $c_j = (L=t, J=f)$   
 $\psi_j(L=t, J=f)$   
 $E_Q[\ln \phi(L, S, J) | L=t, J=f]$   
 $= \sum_s Q(S=s | L=t, J=f) \ln \phi(L=t, S=s, J=f)$

need  $Q(S=s | L=t, J=f)$  ← what is it?  $Q$  is approx of posterior  
 ← what approximation thinks  $P(S=s | L=t, J=f)$  should be  
 compute using variable elimination or clique tree on  $Q = \frac{1}{Z} \prod_i \psi_i(c_i)$

11

# Structured Variational update requires inference

$$\psi_j^{new}(c_j) \propto \exp \left\{ \sum_{\phi \in \mathcal{F}} E_Q^{old}[\ln \phi | c_j] - \sum_{\psi \in \mathcal{Q} \setminus \{\psi_j\}} E_Q^{old}[\ln \psi | c_j] \right\}$$

■ Compute marginals wrt  $Q$  of cliques in original graph and cliques in new graph, for all cliques  
 ■ What is a good way of computing all these marginals? (clique tree)

■ Potential updates:
 

- sequential: compute marginals, update  $\psi_j$ , recompute marginals
- parallel: compute marginals, update all  $\psi$ 's, recompute marginals

12

# What you need to know about variational methods

- Structured Variational method:
  - select a form for approximate distribution  $Q$
  - minimize reverse KL
- Equivalent to maximizing energy functional
  - searching for a tight lower bound on the partition function  $Z$
- Many possible models for  $Q$ :
  - independent (mean field)  $\uparrow \rightarrow Q = \prod_i Q_i(x_i)$
  - structured as a Markov net
  - cluster variational  $\leftarrow Q = \prod_i Q_i(c_i) \quad c_i \cap c_j = \emptyset$
- Several subtleties outlined in the book

10.708 - ©Carlos Guestrin 2006

13

# Announcements

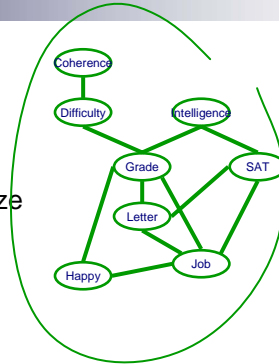
- Tomorrow's recitation
  - Ajit on Loopy BP

10.708 - ©Carlos Guestrin 2006

14

# Recall message passing over junction trees

- Exact inference:
  - generate a junction tree
  - message passing over neighbors
  - inference exponential in size of clique



10.708 - ©Carlos Guestrin 2006

15

# Belief Propagation on Tree Pairwise Markov Nets

- Tree pairwise Markov net is a tree!!! ☺
  - no need to create a junction tree

- Message passing:

$$\delta_{G \rightarrow I}(I) = \sum_g \delta_{D \rightarrow G}(g) \delta_{H \rightarrow G}(g) \delta_{L \rightarrow G}(g) \cdot \phi_G(g) \cdot \phi_{(I, g)}$$

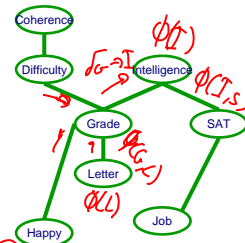
- More general equation:

$$\delta_{i \rightarrow j}(X_j) = \sum_{x_i} \phi_i(x_i) \phi_{ij}(x_i, X_j) \prod_{k \in N(i)-j} \delta_{k \rightarrow i}(x_i)$$

- Theorem: Converges to true probabilities:

$$P(x_i) = \frac{1}{z_i} \phi_i(x_i) \prod_{k \in N(i)} \delta_{k \rightarrow i}(x_i)$$

$$P(X) = \frac{1}{z} \prod_i \phi_i(x_i) \prod_{(i,j) \in \text{edges}} \phi_{ij}(x_i, x_j)$$



Junction tree:



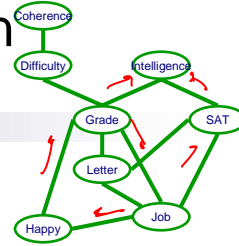
incoming messages other than  $\delta_{j \rightarrow i}$

$$P(x_i, x_j) = \frac{1}{z_{i,j}} \phi_i(x_i) \phi_j(x_j) \phi_{ij}(x_i, x_j) \prod_{k \in N(i)-j} \delta_{k \rightarrow i}(x_i) \prod_{u \in N(j)-i} \delta_{u \rightarrow j}(x_j)$$

10.708 - ©Carlos Guestrin 2006

16

# Loopy Belief Propagation on Pairwise Markov Nets

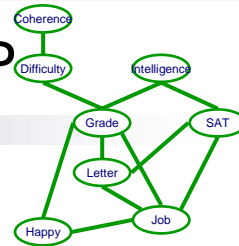


$$\delta_{i \rightarrow j}(X_j) = \sum_{x_i} \phi_i(x_i) \phi_{ij}(x_i, X_j) \prod_{k \in \mathcal{N}(i) - j} \delta_{k \rightarrow i}(x_i)$$

- What if we apply BP in a graph with loops?
  - send messages between pairs of nodes in graph, and hope for the best
- What happens?
  - evidence goes around the loops multiple times
  - may not converge
  - if it converges, usually overconfident about probability values
- But often gives you reasonable, or at least useful answers
  - especially if you just care about the MPE rather than the actual probabilities

MPE  $\Rightarrow \sim \arg \max_x P(x)$   
 MAP  $\Rightarrow X = Y \cup Z \sim \arg \max_y \sum_z P(y, z)$

# More details on Loopy BP



- Numerical problem:
  - messages < 1 get multiplied together as we go around the loops
  - numbers can go to zero
  - normalize messages to one:

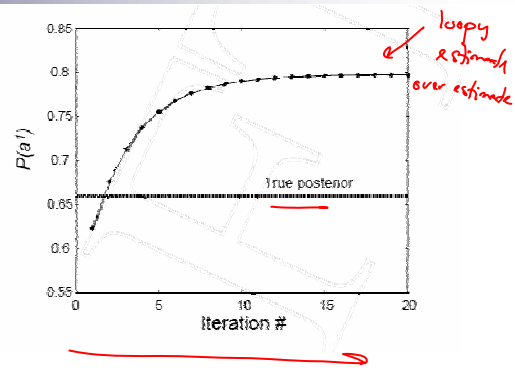
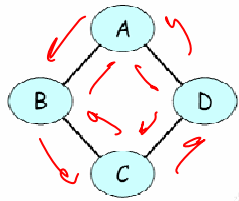
$$\delta_{i \rightarrow j}(X_j) = \frac{1}{Z_{i \rightarrow j}} \sum_{x_i} \phi_i(x_i) \phi_{ij}(x_i, X_j) \prod_{k \in \mathcal{N}(i) - j} \delta_{k \rightarrow i}(x_i)$$

- $Z_{i \rightarrow j}$  doesn't depend on  $X_j$ , so doesn't change the answer

- Computing node "beliefs" (estimates of probs.):

$$\hat{P}(X_i) = \frac{1}{Z_i} \phi_i(X_i) \prod_{k \in \mathcal{N}(i)} \delta_{k \rightarrow i}(X_i)$$

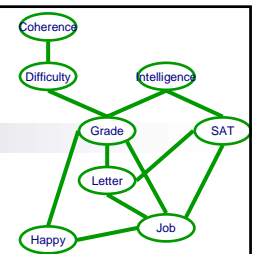
# An example of running loopy BP



usually over confident ...

# Convergence

$$\hat{P}(X_i) = \frac{1}{Z_i} \phi_i(X_i) \prod_{k \in \mathcal{N}(i)} \delta_{k \rightarrow i}(X_i)$$

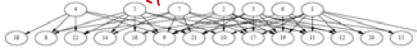
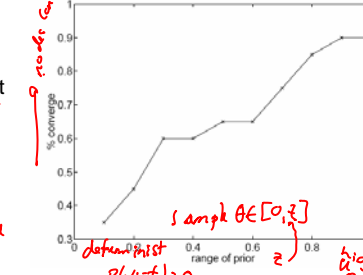
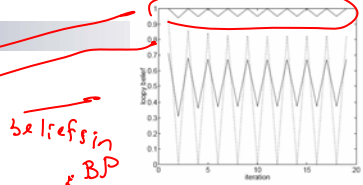


- If you tried to send all messages, and beliefs haven't changed (by much) → converged

normalized!! messages don't change → converged

# (Non-)Convergence of Loopy BP

- **Loopy BP can oscillate!!!**
  - oscillations can be small
  - oscillations can be really bad!
  
- Typically,
  - if factors are closer to uniform, loopy does well (converges)
  - if factors are closer to deterministic, loopy doesn't behave well
  
- One approach to help: damping messages
  - new message is average of old message and new one:  $\lambda \in (0,1)$
  - $$\delta_{i \rightarrow j} = \lambda \delta_{i \rightarrow j}^{\text{last iteration}} + (1-\lambda) \delta_{i \rightarrow j}^{\text{computed}}$$
  - often better convergence
    - but, when damping is required to get convergence, result often bad



graphs from Murphy et al. '99

# Loopy BP in Factor graphs

- What if we don't have pairwise Markov nets?
  1. Transform to a pairwise MN
  2. Use Loopy BP on a factor graph
  
- Message example:
  - from node to factor:
  - from factor to node:



## Loopy BP in Factor graphs

- From node  $i$  to factor  $j$ :

- $F(i)$  factors whose scope includes  $X_i$

$$\delta_{i \rightarrow j}(X_i) \propto \prod_{k \in \mathcal{F}(i) - j} \delta_{k \rightarrow i}(X_i)$$

(A) (B) (C) (D) (E)

ABC ABD BDE CDE

- From factor  $j$  to node  $i$ :

- Scope $[\phi_j] = Y \cup \{X_i\}$

$$\delta_{j \rightarrow i}(X_i) \propto \sum_{\mathbf{y}} \phi_j(X_i, \mathbf{y}) \prod_{X_k \in \text{Scope}[\phi_j] - X_i} \delta_{k \rightarrow j}(x_k)$$

10.708 - ©Carlos Guestrin 2006

23

## What you need to know about loopy BP

- Application of belief propagation in loopy graphs
- Doesn't always converge
  - damping can help
  - good message schedules can help (see book)
- If converges, often to incorrect, but useful results
- Generalizes from pairwise Markov networks by using factor graphs

10.708 - ©Carlos Guestrin 2006

24