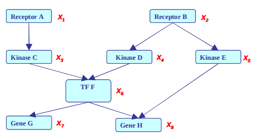
School of Computer Science
Carnegie Mellon

Approximate Inference: Loopy Belief Propagation and variants

Probabilistic Graphical Models (10-708)

Lecture 14, Nov 7, 2007

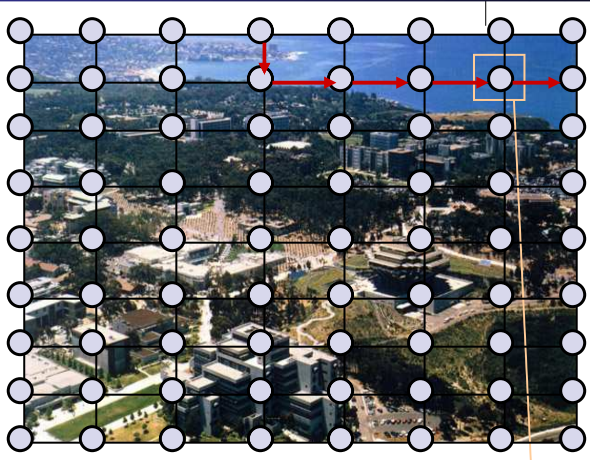


Hetunandan Kamisetty

Reading: J-Chap. 5,6, KF-Chap. 8

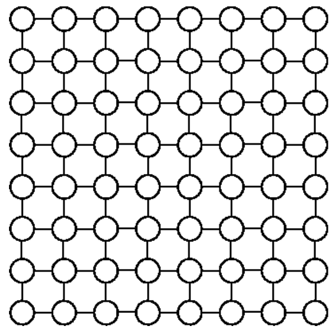
An Ising model on 2-D image

- Nodes encode hidden information (patch-identity).
- They receive local information from the image (brightness, color).
- Information is propagated though the graph over its edges.
- Edges encode 'compatibility' between nodes.



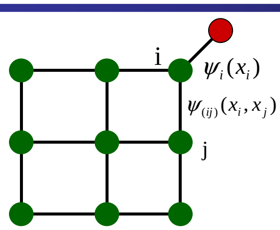
air or water ?

Why Approximate Inference?



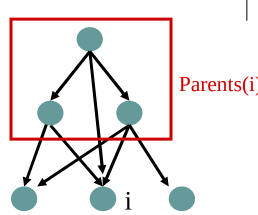
- Why can't we just run junction tree on this graph?
- If NxN grid, tree width atleast N
  - If N~O(1000), we have a clique with 2^{100} entries

A recap: Factor Graphs



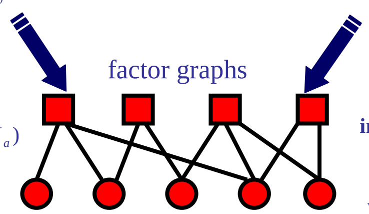
Undirected graph
(Markov random field)

$P(x) = \frac{1}{Z}\prod_i \psi_i(x_i) \prod_{(ij)} \psi_{(ij)}(x_i, x_j)$



Directed graph
(Bayesian network)

$P(x) = \prod_i P(x_i \mid x_{\text{parents}(i)})$

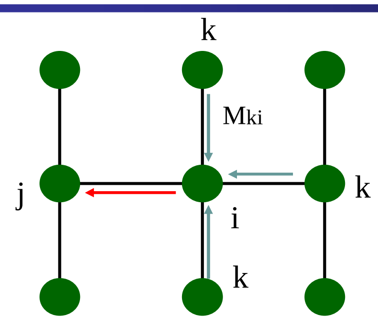


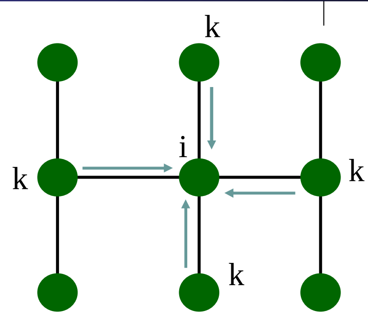
$P(X) = 1/Z \prod_{f_a \in F} f_a(X_a)$

interactions $f_a(X_a)$

variables

Belief Propagation





- BP Message-update Rules

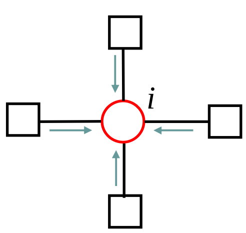
$$M_{i\to j}(x_j) \propto \sum_{x_i} \psi_{ij}(x_i, x_j)\psi_i(x_i)\prod_k M_{k\to i}(x_i) \qquad b_i(x_i) \propto \psi_i(x_i)\prod_k M_k(x_k)$$

external evidence

Compatibilities (interactions)

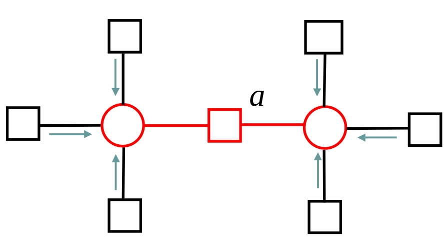
- BP on trees always converges to exact marginals (cf. Junction tree algorithm)

Beliefs and messages in FG



$$b_i(x_i) \propto f_i(x_i)\prod_{a\in N(i)} m_{a\to i}(x_i)$$

"beliefs" "messages"



$$m_{i\to a}(x_i) = \prod_{c\in N(i)\backslash a} m_{c\to i}(x_i)$$

$$b_a(X_a) \propto f_a(X_a)\prod_{i\in N(a)} m_{i\to a}(x_i)$$

$$m_{a\to i}(x_i) = \sum_{X_a\backslash x_i} f_a(X_a)\prod_{j\in N(a)\backslash i} m_{j\to a}(x_j)$$

Approximate Inference: What to approximate?

- Let us call the actual distribution P

$$P(X) = 1/Z \prod_{f_a \in F} f_a(X_a)$$

- We wish to find a distribution Q such that Q is a "good" approximation
- Recall the definition of KL-divergence

$$KL(Q_1 \| Q_2) = \sum_X Q_1(X) \log(\frac{Q_1(X)}{Q_2(X)})$$

  - $KL(Q_1||Q_2)>=0$
  - $KL(Q_1||Q_2)=0$ iff $Q_1=Q_2$
  - We can therefore use KL as a scoring function to decide a good Q
  - But, $KL(Q_1||Q_2) \neq KL(Q_2||Q_1)$

Which KL?

- Computing KL(P||Q) requires inference!
- But KL(Q||P) can be computed without performing inference on P

$$KL(Q \| P) = \sum_X Q(X) \log(\frac{Q(X)}{P(X)})$$
$$= \sum_X Q(X) \log Q(X) - \sum_X Q(X) \log P(X)$$
$$= -H_Q(X) - E_Q \log P(X)$$

- Using $P(X) = 1/Z \prod_{f_a \in F} f_a(X_a)$

$$KL(Q \| P) = -H_Q(X) - E_Q \log(1/Z \prod_{f_a \in F} f_a(X_a))$$
$$= -H_Q(X) - \log 1/Z - \sum_{f_a \in F} E_Q \log f_a(X_a)$$

Optimization function

$$KL(Q \| P) = \underbrace{\boxed{-H_Q(X) - \sum_{f_a \in F} E_Q \log f_a(X_a)}}_{F(P,Q)} + \log Z$$

- We will call $F(P,Q)$ the "Free energy" *
- $F(P,P)$ =?
- F(P,Q) >= F(P,P)

*Gibbs Free Energy

The Free Energy

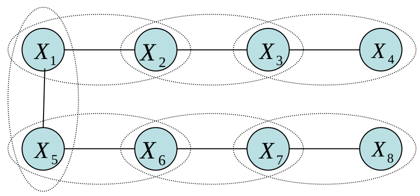
- Let us look at the free energy

$$F(P,Q) = -H_Q(X) - \sum_{f_a \in F} E_Q \log f_a(X_a)$$

- $\sum_{f_a \in F} E_Q \log f_a(X_a)$ can be computed if we have marginals over each f_a
- $H_Q = \sum_X Q(X) \log Q(X)$ is harder! Requires summation over all possible values
- Computing F, is therefore hard in general.
- Approach 1: Approximate $F(P,Q)$ with easy to compute $\hat{F}(P,Q)$

Easy free energies

- Consider a tree-structured distribution



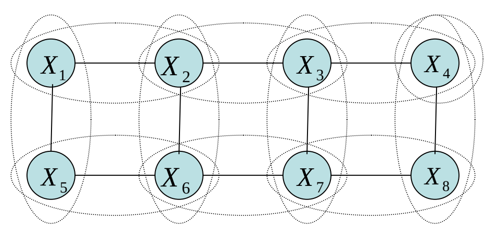
- The probability can be written as: $b(\mathbf{x}) = \prod_a b_a(\mathbf{x}_a) \prod_i b_i(x_i)^{1-d_i}$
- $H_{tree} = -\sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \log b_a(\mathbf{x}_a) + \sum_i (d_i - 1) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \log b_i(\mathbf{x}_i)$
- $F_{Tree} = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \log \frac{b_a(\mathbf{x}_a)}{f_a(\mathbf{x}_a)} + \sum_i (1 - d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \log b_i(\mathbf{x}_i)$

  $= F_{12} + F_{23} + .. + F_{67} + F_{78} - F_1 - F_5 - F_2 - F_6 - F_3 - F_7$
  - involves summation over edges and vertices and is therefore easy to compute

Bethe Approximation to Gibbs Free Energy

- For a general graph, choose

  $$\hat{F}(P,Q) = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \log \frac{b_a(\mathbf{x}_a)}{f_a(\mathbf{x}_a)} + \sum_i (1 - d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \log b_i(\mathbf{x}_i)$$
  - Called “Bethe approximation” after the physicist Hans Bethe
  - Equal to the exact Gibbs free energy when the factor graph is a tree
  - Note: This is **not** the same as the entropy of a tree



$$F_{bethe} = F_{12} + F_{23} + .. + F_{67} + F_{78} - F_1 - F_5 - 2F_2 - 2F_6 .. - F_8$$

Bethe Approximation

- Pros:
  - Easy to compute, since entropy term involves sum over pairwise and single variables
- Cons:
  - $\hat{F}(P,Q) = F_{bethe}$ **may or may not** be well connected to $F(P,Q)$
  - It could, in general, be greater, equal or less than $F(P,Q)$
- Optimize each $b(\mathbf{x}_a)$'s.
  - For discrete belief, constrained opt. with *Lagrangian* multiplier
  - For continuous belief, not yet a general formula
  - Not always converge

Minimizing the Bethe Free Energy

- $$L = F_{Bethe} + \sum_i \gamma_i \{1 - \sum_{x_i} b_i(x_i)\}$$
  $$+ \sum_a \sum_{i \in N(a)} \sum_{x_i} \lambda_{ai}(x_i) \left\{ b_i(x_i) - \sum_{X_a \backslash x_i} b_a(X_a) \right\}$$
- Set derivative to zero

Proof

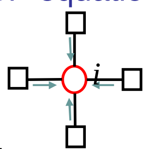
-

Bethe = BP

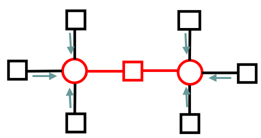
- We had

$$b_i(x_i) \propto \exp\left(\frac{1}{d_i - 1}\sum_{a \in N(i)} \lambda_{ai}(x_i)\right) \qquad b_a(X_a) \propto \exp\left(-\log f_a(X_a) + \sum_{i \in N(a)} \lambda_{ai}(x_i)\right)$$

- Identify $\lambda_{ai}(x_i) = \log(m_{i \to a}(x_i)) = \log \prod_{b \in N(i) \neq a} m_{b \to i}(x_i)$
- to obtain BP equations:



$$b_i(x_i) \propto \prod_{a \in N(i)} m_{a \to i}(x_i)$$



$$b_a(X_a) \propto f_a(X_a) \prod_{i \in N(a)} \prod_{b \in N(i) \setminus a} m_{b \to i}(x_i)$$

Loopy Belief Propagation

- A fixed point iteration procedure that tries to minimize F_{bethe}
- Start with random initialization of messages and beliefs
  - While not converged do

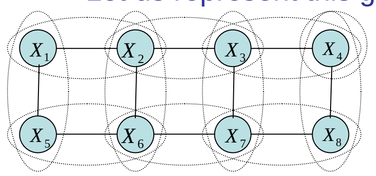
$$b_i(x_i) \propto \prod_{a \in N(i)} m_{a \to i}(x_i) \qquad b_a(X_a) \propto f_a(X_a) \prod_{i \in N(a)} m_{i \to a}(x_i)$$

$$m_{i \to a}^{new}(x_i) = \prod_{c \in N(i) \setminus a} m_{c \to i}(x_i) \qquad m_{a \to i}^{new}(x_i) = \sum_{X_a \setminus x_i} f_a(X_a) \prod_{j \in N(a) \setminus i} m_{j \to a}(x_j)$$

  - At convergence, stationarity properties are guaranteed
  - However, not guaranteed to converge!

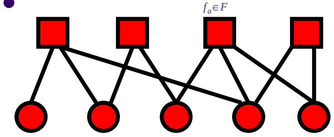
Region graphs

- It will be useful to look explicitly at the messages being passed
  - Messages from variable to factors
  - Messages from factors to variables
- Let us represent this graphically



Summary so far

- $P(X) = 1/Z \prod_{f_a \in F} f_a(X_a)$

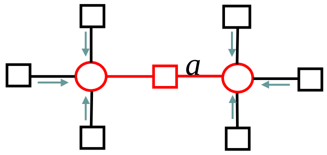


$$F(P,Q) = -H_Q(X) - \sum_{f_a \in F} E_Q \log f_a(X_a)$$

$$\hat{F}(P,Q) = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \log \frac{f_a(\mathbf{x}_a)}{b_a(\mathbf{x}_a)} + \sum_i (1-d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \log b_i(\mathbf{x}_i)$$

$$b_a(X_a) \propto \exp\left(-\log f_a(X_a) + \sum_{i \in N(a)} \lambda_{ai}(x_i) \right)$$

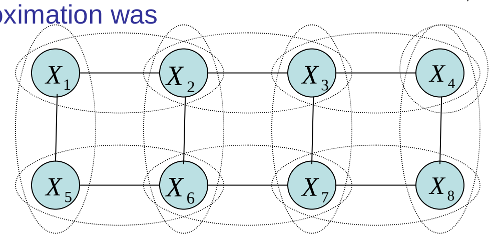
$$b_i(x_i) \propto \exp\left(\frac{1}{d_i - 1} \sum_{a \in N(i)} \lambda_{ai}(x_i) \right)$$



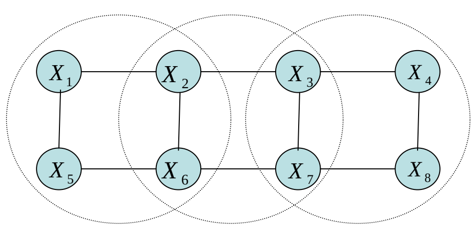
Better approximations?

- Recall that Bethe approximation was

$$F_{bethe} = F_{12} + F_{23} + .. + F_{67} + F_{78} - F_1 - F_5 - 2F_2 - 2F_6 .. - F_8$$



- We could construct bigger regions
- Rules:
  - If a region includes a factor, it must include the vertices as well
  - Each factor and vertex must appear in atleast one region
  - Associate a weight with each region so that each vertex and factor is counted exactly once



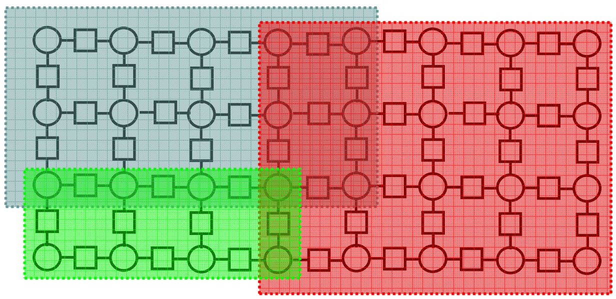
$$\hat{F} = F_{1256} + F_{2367} + F_{3478} - F_{26} - F_{36} + F_2 + F_6 + F_3 + F_7$$

- Other regions?

Region-based Approximations to the Gibbs Free Energy (Kikuchi, 1951)

Exact: $G[q(X)]$ *(intractable)*

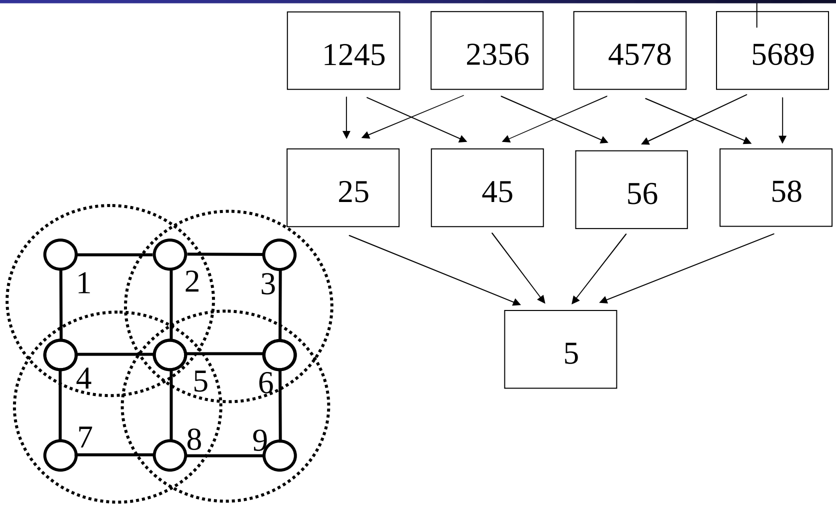
Regions: $G[\{b_r(X_r)\}]$



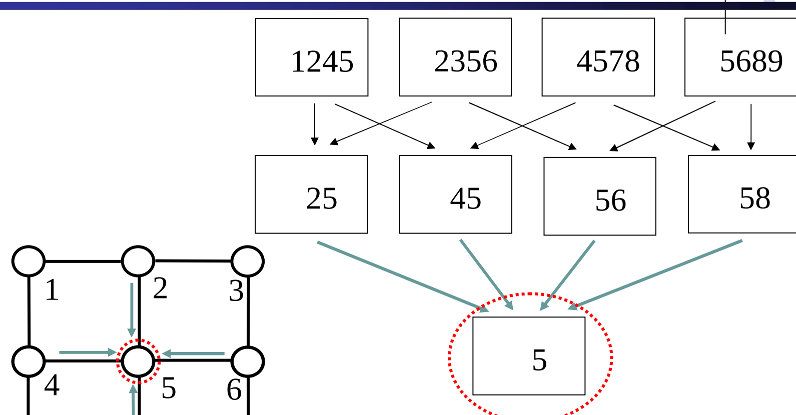
Generalized Belief Propagation

- Belief in a region is the product of:
  - Local information (factors in region)
  - Messages from parent regions
  - Messages into descendant regions from parents who are not descendants.
- Message-update rules obtained by enforcing marginalization constraints.

Generalized Belief Propagation

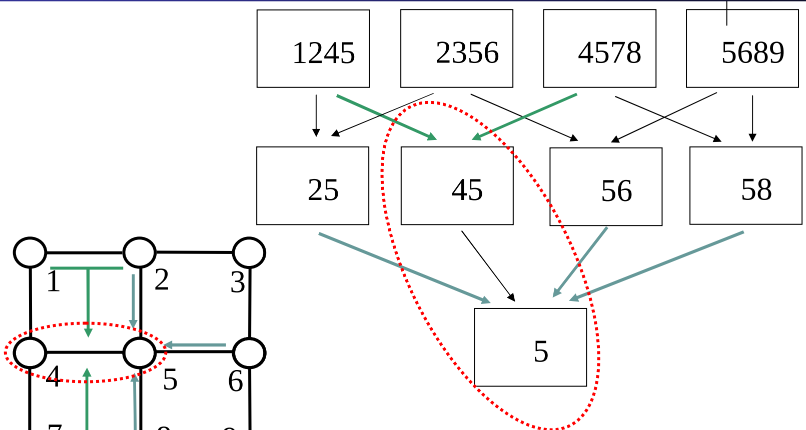


Generalized Belief Propagation



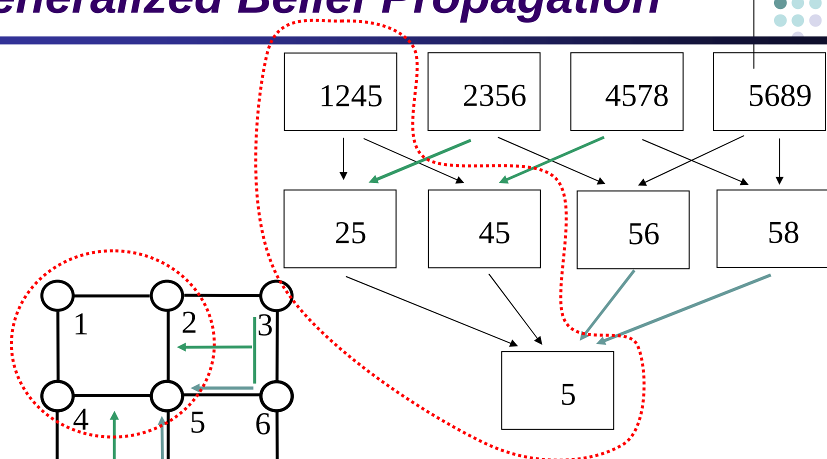
$$b_5 \propto m_{2\to5} m_{4\to5} m_{6\to5} m_{8\to5}$$

Generalized Belief Propagation



$$b_{45} \propto [\, f_{45}][m_{12\to45} m_{78\to45} m_{2\to5} m_{6\to5} m_{8\to5}]$$

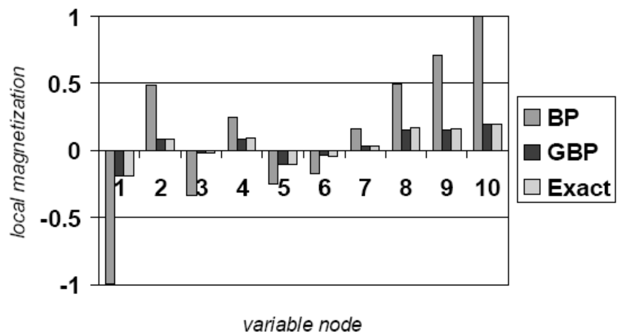
Generalized Belief Propagation



$$b_{1245} \propto [\, f_{12}\, f_{14}\, f_{25}\, f_{45}][m_{36\to25} m_{78\to45} m_{6\to5} m_{8\to5}]$$

Some results

•



Summary

- We defined an objective function (F) for approximate inference
- However, we found that optimizing this function was hard
- We first approximated objective function F to simpler F_{bethe}
  - Minima of F_{bethe} turned out to be fixed points of BP
- Then we extended this to more complicated approximations
  - The resulting algorithms come under a family called Generalized Belief Propagation
- Next class, we will cover other methods of approximations