Probabilistic Graphical Models

Variational Inference:
Loopy Belief Propagation

Eric Xing
Lecture 12, February 23, 2015
Reading: See class website
Inference Problems

- Compute the likelihood of observed data
- Compute the marginal distribution $p(x_A)$ over a particular subset of nodes $A \subset V$
- Compute the conditional distribution $p(x_A|x_B)$ for disjoint subsets $A$ and $B$
- Compute a mode of the density $\hat{x} = \arg \max_{x \in X^m} p(x)$

- Methods we have
  - Brute force
  - Elimination
  - Message Passing
    - (Forward-backward, Max-product /BP, Junction Tree)
  - Individual computations independent
  - Sharing intermediate terms
Sum-Product Revisited

- Tree-structured GMs

\[ p(x_1, \ldots, x_m) = \frac{1}{Z} \prod_{s \in V} \psi_s(x_s) \prod_{(s, t) \in E} \psi_{st}(x_s, x_t) \]

- Message Passing on Trees:

\[ M_{t \rightarrow s}(x_s) \leftarrow \kappa \sum_{x'_t} \left\{ \psi_{st}(x_s, x'_t) \psi_{t}(x'_t) \prod_{u \in N(t) \setminus s} M_{u \rightarrow t}(x'_t) \right\} \]

- On trees, converge to a unique fixed point after a finite number of iterations
Junction Tree Revisited

- General Algorithm on Graphs with Cycles

- Steps:
  - Triangularization
  - Construct JTs

=> Message Passing on Clique Trees

\[
\tilde{\phi}_S(x_S) \leftarrow \sum_{x_{B \setminus S}} \phi_B(x_B)
\]

\[
\phi_C(x_C) \leftarrow \frac{\tilde{\phi}_S(x_S)}{\phi_S(x_S)} \phi_C(x_C)
\]
Local Consistency

- Given a set of functions $\{\tau_C, \ C \in \mathcal{C}\}$ and $\{\tau_S, \ S \in \mathcal{S}\}$ associated with the cliques and separator sets

- They are locally consistent if:

  $$\sum_{x'_S} \tau_S(x'_S) = 1, \ \forall S \in \mathcal{S}$$

  $$\sum_{x'_C | x'_C = x_S} \tau_C(x'_C) = \tau_S(x_S), \ \forall C \in \mathcal{C}, \ S \subset C$$

- For junction trees, local consistency is equivalent to global consistency!
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.

© Eric Xing @ CMU, 2005-2015
Why Approximate Inference?

- Why can’t we just run junction tree on this graph?

- If N x N grid, tree width at least N
- N can be a huge number (~1000s of pixels)
  - If N ~ O(1000), we have a clique with $2^{100}$ entries

$$p(X) = \frac{1}{Z} \exp\left\{ \sum_{i < j} \theta_{ij} X_i X_j + \sum_i \theta_{i0} X_i \right\}$$
Approaches to inference

- Exact inference algorithms
  - The elimination algorithm
  - Message-passing algorithm (sum-product, belief propagation)
  - The junction tree algorithms

- Approximate inference techniques
  - Variational algorithms
    - Loopy belief propagation ✓
    - Mean field approximation ✓
  - Stochastic simulation / sampling methods ✓
  - Markov chain Monte Carlo methods ✓
Loopy Belief Propagation
Recap: Belief Propagation

- **BP Message-update Rules**
  
  \[ M_{i \rightarrow j}(x_j) \propto \sum_{x_i} \psi_{ij}(x_i, x_j) \psi_i(x_i) \prod_k M_{k \rightarrow i}(x_i) \]
  
  - Compatibilities (interactions)
  
  \[ b_i(x_i) \propto \psi_i(x_i) \prod_k M_k(x_k) \]
  
  - External evidence

- **BP on trees always converges to exact marginals (cf. Junction tree algorithm)**
Region graphs (Factor Graph)

- 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

![Diagram of Region Graphs]

© Eric Xing @ CMU, 2005-2015
Beliefs and messages in FG

\[ b_i(x_i) \propto f_i(x_i) \prod_{a \in N(i)} m_{a \rightarrow i}(x_i) \]

“beliefs”

\[ m_{i \rightarrow a}(x_i) = \prod_{c \in N(i) \setminus a} m_{c \rightarrow i}(x_i) \]

\[ b_a(X_a) \propto f_a(X_a) \prod_{i \in N(a)} m_{i \rightarrow a}(x_i) \]

\[ m_{a \rightarrow i}(x_i) = \sum_{X_a \setminus x_i} f_a(X_a) \prod_{j \in N(a) \setminus i} m_{j \rightarrow a}(x_j) \]
What if the graph is loopy?
Belief Propagation on loopy graphs

- BP Message-update Rules

\[ M_{i \rightarrow j}(x_j) \propto \sum_{x_i} \psi_{ij}(x_i, x_j) \psi_i(x_i) \prod_k M_{k \rightarrow i}(x_i) \]

\[ b_j(x_j) \propto \psi_j(x_j) \prod_k M_k(x_k) \]

- May not converge or converge to a wrong solution
Loopy Belief Propagation

- A fixed point iteration procedure that tries to minimize $F_{\text{bethe}}$
- Start with random initialization of messages and beliefs
  - While not converged do
    \[
    b_i(x_i) \propto \prod_{a \in N(i)} m_{a\rightarrow i}(x_i) \quad b_a(X_a) \propto f_a(X_a) \prod_{i \in N(a)} m_{i\rightarrow a}(x_i)
    \]
    \[
    m_{i\rightarrow a}^{\text{new}}(x_i) = \prod_{c \in N(i) \setminus a} m_{c\rightarrow i}(x_i) \quad m_{a\rightarrow i}^{\text{new}}(x_i) = \sum_{X_a \setminus x_i} f_a(X_a) \prod_{j \in N(a) \setminus i} m_{j\rightarrow a}(x_j)
    \]
  - At convergence, stationarity properties are guaranteed
  - However, not guaranteed to converge!
Loopy Belief Propagation

- If BP is used on graphs with loops, messages may circulate indefinitely

- But let’s run it anyway and hope for the best … 😊

- Empirically, a good approximation is still achievable
  - Stop after fixed # of iterations
  - Stop when no significant change in beliefs
  - If solution is not oscillatory but converges, it usually is a good approximation

---

Loopy-belief Propagation for Approximate Inference: An Empirical Study
Kevin Murphy, Yair Weiss, and Michael Jordan.
UAI ’99 (Uncertainty in AI).

© Eric Xing @ CMU, 2005-2015
So what is going on?

- Is it a dirty hack that you bet your luck?
Approximate Inference

- Let us call the actual distribution $P$

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

- We wish to find a distribution $Q$ such that $Q$ is a “good” approximation to $P$

- Recall the definition of KL-divergence

$$KL(Q_1 \parallel Q_2) = \sum_X Q_1(X) \log \left( \frac{Q_1(X)}{Q_2(X)} \right)$$

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

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

\[
KL(Q \| \| P) = \sum_{X} Q(X) \log \left( \frac{Q(X)}{P(X)} \right)
\]

\[
= \sum_{X} Q(X) \log Q(X) - \sum_{X} Q(X) \log P(X)
\]

\[
= -H_{Q}(X) - E_{Q} \log P(X)
\]

- Using

\[
P(X) = \frac{1}{Z} \prod_{f_{a} \in F} f_{a}(X_{a})
\]

\[
KL(Q \| \| P) = -H_{Q}(X) - E_{Q} \log \left( \frac{1}{Z} \prod_{f_{a} \in F} f_{a}(X_{a}) \right)
\]

\[
= -H_{Q}(X) - \log 1/Z - \sum_{f_{a} \in F} E_{Q} \log f_{a}(X_{a})
\]

$P(x) = \frac{1}{Z_{T}} e^{\theta^{T}(x)}$

$Q(x) = \Pi q(x_{i})$
Optimization function

$$\text{KL}(Q \parallel P) = -H_Q(X) - \sum_{f_a \in F} E_Q \log f_a(X_a) + \log Z$$

$$F(P,Q)$$

- We will call $F(P,Q)$ the “Free energy” *
- $F(P,P) = -\log \mathcal{Z}$
- $F(P,Q) \geq F(P,P)$

*Gibbs Free Energy
Let us look at the functional

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

\[ \sum_{f_a \in F} E_Q \log f_a(X_a) \]

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 \( \hat{F}(P, Q) \)

\[ Q^* = \arg \max \hat{F}. \]

\[ F \] is hard! \( \hat{F} \)

\[ Q^* = \arg \max \hat{F} \]
Tree Energy Functionals

- Consider a tree-structured distribution

\[ p(x_1, \ldots, x_8) = p(x_8) p(x_1, \ldots, x_7) \]

\[ p(x_6, x_5) p(x_5, x_4) p(x_4, x_3) p(x_3, x_2) p(x_2, x_1) \]

- The probability can be written as:

\[ b(x) = \prod_{a} b_a(x_a) \prod_{i} b_i(x_i)^{1-d_i} \]

- \[ H_{\text{tree}} = -\sum_{a} \sum_{x_a} b_a(x_a) \ln b_a(x_a) + \sum_{i} (d_i - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i) \]

- \[ F_{\text{Tree}} = \sum_{a} \sum_{x_a} b_a(x_a) \ln \frac{b_a(x_a)}{f_a(x_a)} + \sum_{i} (1 - d_i) \sum_{x_i} b_i(x_i) \ln b_i(x_i) \]

\[ = F_{12} + F_{23} + \ldots + F_{67} + F_{78} - F_1 - F_5 - F_2 - F_6 - F_3 - F_7 \]

- involves summation over edges and vertices and is therefore easy to compute

© Eric Xing @ CMU, 2005-2015
Bethe Approximation to Gibbs Free Energy

- For a general graph, choose $F(P,Q) = F_{\text{Bethe}}$

$H_{\text{Bethe}} = - \sum_a \sum_{x_a} b_a(x_a) \ln b_a(x_a) + \sum_i (d_i - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i)$

$F_{\text{Bethe}} = \sum_a \sum_{x_a} b_a(x_a) \ln \frac{b_a(x_a)}{f_a(x_a)} + \sum_i (1 - d_i) \sum_{x_i} b_i(x_i) \ln b_i(x_i) = -\langle f_a(x_a) \rangle - H_{\text{Bethe}}$

- Called “Bethe approximation” after the physicist Hans Bethe

$F_{\text{Bethe}} = F_{12} + F_{23} + \ldots + F_{67} + F_{78} - F_1 - F_5 - 2F_2 - 2F_6 \ldots - F_8$

- Equal to the exact Gibbs free energy when the factor graph is a tree
- In general, $H_{\text{Bethe}}$ is not the same as the $H$ of a tree

© Eric Xing @ CMU, 2005-2015
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** \( b(x_a)'s. \)
  - For discrete belief, constrained opt. with *Lagrangian* multiplier
  - For continuous belief, not yet a general formula
  - Not always converge

\[
F_{bethe} \neq F_a
\]
Bethe Free Energy for FG

\[ F_{\text{Bethe}} = \sum_a \sum_{x_a} p_a(x_a) \ln \frac{b_a(x_a)}{f_a(x_a)} + \sum_i (1 - d_i) \sum_{x_i} b_i(x_i) \ln b_i(x_i) \]

\[ H_{\text{Bethe}} = -\sum_a \sum_{x_a} b_a(x_a) \ln b_a(x_a) + \sum_i (d_i - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i) \]

\[ F_{\text{Bethe}} = -\langle f_a(x_a) \rangle - H_{\text{Bethe}} \]

\[ \sum_{x_i} b_i(x_i) = 1 \]

\[ \sum_{x_{\alpha}} b_{\alpha}(x_{\alpha}) = 1 \]

\[ \sum_{x_{\alpha}} (b_{\alpha}(x_{\alpha}) = b(x_i) \]
Minimizing the Bethe Free Energy

\[ L = F_{\text{Bethe}} + \sum_i \gamma_i \{ 1 - \sum_x b_i(x_i) \} \]

\[ + \sum_a \sum_{i \in N(a)} \sum_x \lambda_{ai}(x_i) \left\{ b_i(x_i) - \sum_{X_a \setminus x} b_a(X_a) \right\} \]

- Set derivative to zero

\[ \frac{\partial L}{\partial b_i} = 0 \]

\[ \frac{\partial L}{\partial b_a} = 0 \]
Constrained Minimization of the Bethe Free Energy

\[
L = F_{\text{Bethe}} + \sum_i \gamma_i \left\{ \sum_{x_i} b_i(x_i) - 1 \right\}
\]

\[
+ \sum_a \sum_{i \in N(a)} \sum_{x_i} \lambda_{ai}(x_i) \left\{ \sum_{X_a \setminus x_i} b_a(X_a) - b_i(x_i) \right\}
\]

\[
\frac{\partial L}{\partial b_i(x_i)} = 0 \quad \iff \quad b_i(x_i) \propto \exp \left( \frac{1}{d_i - 1} \sum_{a \in N(i)} \lambda_{ai}(x_i) \right)
\]

\[
\frac{\partial L}{\partial b_a(X_a)} = 0 \quad \iff \quad b_a(X_a) \propto \exp \left( -E_a(X_a) + \sum_{i \in N(a)} \lambda_{ai}(x_i) \right)
\]

© Eric Xing @ CMU, 2005-2015 27
Bethe = BP on FG

- We had:
  \[ b_i(x_i) \propto \exp \left( \frac{1}{d_i - 1} \sum_{a \in N(i)} \lambda_{ai}(x_i) \right) \]
  \[ b_a(X_a) \propto \exp \left( -\log f_a(X_a) + \sum_{i \in N(a)} \lambda_{ai}(x_i) \right) \]

- Identify \( \lambda_{ai}(x_i) = \log(m_{i \rightarrow a}(x_i)) = \log \prod_{b \in N(i) \neq a} m_{b \rightarrow i}(x_i) \)

- to obtain BP equations:

The “belief” is the BP approximation of the marginal probability.
BP Message-update Rules

Using \( b_{a \rightarrow i}(x_i) = \sum_{X_a \setminus x_i} b_a(X_a) \), we get

\[
m_{a \rightarrow i}(x_i) = \sum_{X_a \setminus x_i} f_a(X_a) \prod_{j \in N(a) \setminus i} \prod_{b \in N(j) \setminus a} m_{b \rightarrow j}(x_j)
\]

( A sum product algorithm )
Summary so far

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

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

\[
\hat{F}(P, Q) = \sum_a \sum_{x_a} b_a(x_a) \log \frac{f_a(x_a)}{b_a(x_a)} + \sum_i (1-d_i) \sum_{x_i} b_i(x_i) \log b_i(x_i)
\]

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

\[
b_i(x_i) \propto \exp \left( \frac{1}{d_i-1} \sum_{a \in N(i)} \lambda_{ai}(x_i) \right)
\]
For a distribution $p(X|\theta)$ associated with a complex graph, computing the marginal (or conditional) probability of arbitrary random variable(s) is intractable.

Variational methods

formulating probabilistic inference as an optimization problem:

$$q^* = \arg\min_{q \in \mathcal{S}} \left\{ F_{\text{Bethe}}(p,q) \right\}$$

$$F_{\text{Bethe}} = \sum_a \sum_{x_a} b_a(x_a) \ln \frac{b_a(x_a)}{f_a(x_a)} + \sum_i (1-d_i) \sum_{x_i} b_i(x_i) \ln b_i(x_i) = -\langle f_a(x_a) \rangle - H_{\text{bethe}}$$

$q$: a (tractable) probability distribution
The Theory Behind LBP

- But we do not optimize $q(X)$ explicitly, focus on the set of beliefs

  - e.g., $b = \{b_{i,j} = \tau(x_i, x_j), \ b_i = \tau(x_i)\}$

- Relax the optimization problem

  - approximate objective: $H_q \approx F(b)$
  - relaxed feasible set: $\mathcal{M} \rightarrow \mathcal{M}_o \ (\mathcal{M}_o \supseteq \mathcal{M})$

  $$b^* = \arg \min_{b \in \mathcal{M}_o} \left\{ \langle E \rangle_b + F(b) \right\}$$

- The loopy BP algorithm:
  - a fixed point iteration procedure that tries to solve $b^*$
The Theory Behind LBP

- But we do not optimize \( q(X) \) explicitly, focus on the set of beliefs
  - \( e.g., \quad b = \{ b_{i,j} = \tau(x_i, x_j), \quad b_i = \tau(x_i) \} \)

- Relax the optimization problem
  - approximate objective: \( H_{Betha} = H(b_{i,j}, b_i) \)
  - relaxed feasible set: \( \mathcal{M}_o = \{ \tau \geq 0 \mid \sum_{x_i} \tau(x_i) = 1, \sum_{x_i} \tau(x_i, x_j) = \tau(x_j) \} \)

\[
\begin{align*}
b^* &= \arg \min_{b \in \mathcal{M}_o} \left\{ \langle E \rangle_b + F(b) \right\}
\end{align*}
\]

- The loopy BP algorithm:
  - a fixed point iteration procedure that tries to solve \( b^* \)
Region-based Approximations to the Gibbs Free Energy (Kikuchi, 1951)

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

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

\[ b_5 \propto m_{2 \rightarrow 5} m_{4 \rightarrow 5} m_{6 \rightarrow 5} m_{8 \rightarrow 5} \]
Generalized Belief Propagation

\[ b_{45} \propto [f_{45}][m_{12 \rightarrow 45}m_{78 \rightarrow 45}m_{2 \rightarrow 5}m_{6 \rightarrow 5}m_{8 \rightarrow 5}] \]
Generalized Belief Propagation

\[ b_{1245} \propto [f_{12}, f_{14}, f_{25}, f_{45}][m_{36 \rightarrow 25} m_{78 \rightarrow 45} m_{6 \rightarrow 5} m_{8 \rightarrow 5}] \]
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 \( F_{\text{bethe}} \):
  - Minima of \( F_{\text{bethe}} \) 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.
Mean Field Approximation
Naïve Mean Field

- Fully factorized variational distribution

\[ q(x) = \prod_{s \in V} q(x_s) \]
Naïve Mean Field for Ising Model

- Optimization Problem

\[
\max_{\mu \in [0,1]^m} \left\{ \sum_{s \in V} \theta_s \mu_s + \sum_{(s,t) \in E} \theta_{st} \mu_s \mu_t + \sum_{s \in V} H_s(\mu_s) \right\}
\]

- Update Rule

\[
\mu_s \leftarrow \sigma \left( \theta_s + \sum_{t \in N(s)} \theta_{st} \mu_t \right)
\]

- \(\mu_t = p(X_t = 1) = \mathbb{E}_p[X_t]\) resembles “message” sent from node \(t\) to \(s\)

- \(\{\mathbb{E}_p[X_t], t \in N(s)\}\) forms the “mean field” applied to \(s\) from its neighborhood
**Mean field methods**

- Optimize $q(X_{H})$ in the space of tractable families
  - *i.e.*, subgraph of $G_p$ over which exact computation of $H_q$ is feasible

- Tightening the optimization space
  - exact objective: $H_q$
  - tightened feasible set: $Q \rightarrow T$ ($T \subseteq Q$)

\[ q^* = \arg\min_{q \in T} \langle E \rangle_q - H_q \]
Cluster-based approx. to the Gibbs free energy (Wiegerinck 2001, Xing et al 03,04)

Exact: $G[p(X)]$  (intractable)

Clusters: $G[\{q_c(X_c)\}]$
Mean field approx. to Gibbs free energy

- Given a disjoint clustering, \{C_1, \ldots, C_I\}, of all variables
- Let
  \[ q(X) = \prod_i q_i(X_{C_i}) , \]
- Mean-field free energy
  \[ G_{MF} = \sum_i \sum_{x_{C_i}} \prod_i q_i(x_{C_i}) E(x_{C_i}) + \sum_i \sum_{x_{C_i}} q_i(x_{C_i}) \ln q_i(x_{C_i}) \]
  e.g.,
  \[ G_{MF} = \sum_{i\neq j} \sum_{x_i x_j} q(x_i)q(x_j) \phi(x_i, x_j) + \sum_i \sum_{x_i} q(x_i) \phi(x_i) + \sum_i \sum_{x_i} q(x_i) \ln q(x_i) \] (naïve mean field)

- Will **never** equal to the exact Gibbs free energy no matter what clustering is used, but it does **always** define a lower bound of the likelihood

- **Optimize each** \( q_i(x_{C_i}) \)'s.
  - Variational calculus …
  - Do inference in each \( q_i(x_{C_i}) \) using any tractable algorithm
**Theorem:** The optimum GMF approximation to the cluster marginal is isomorphic to the cluster posterior of the original distribution given internal evidence and its generalized mean fields:

\[ q_i^*(X_{H,C_i}) = p(X_{H,C_i} \mid X_{E,C_i}, \left< X_{H,MB_i} \right>_{q_{j\neq i}}) \]

**GMF algorithm:** Iterate over each \( q_i \)
A generalized mean field algorithm [xing et al. UAI 2003]
A generalized mean field algorithm [xing et al. UAI 2003]


Theorem: The GMF algorithm is guaranteed to converge to a local optimum, and provides a lower bound for the likelihood of evidence (or partition function) the model.
The naive mean field approximation

- Approximate $p(X)$ by fully factorized $q(X) = \prod_i q_i(X_i)$

- For Boltzmann distribution $p(X) = \exp \left\{ \sum_{i<j} q_{ij} X_i X_j + q_{io} X_i \right\} / Z$

  mean field equation:

  $q_i(X_i) = \exp \left\{ \theta_{i0} X_i + \sum_{j \in \mathcal{N}_i} \theta_{ij} X_i \left\langle X_j \right\rangle_{q_j} + A_i \right\}$

  $= p(X_i | \{\left\langle X_j \right\rangle_{q_j} : j \in \mathcal{N}_i \})$

  - $\left\langle X_j \right\rangle_{q_j}$ resembles a “message” sent from node $j$ to $i$
  - $\{\left\langle X_j \right\rangle_{q_j} : j \in \mathcal{N}_i \}$ forms the “mean field” applied to $X_i$ from its neighborhood
Example 1: Generalized MF approximations to Ising models

Cluster marginal of a square block $C_k$:

$$q(X_{C_k}) \propto \exp \left\{ \sum_{i,j \in C_k} \theta_{ij} X_i X_j + \sum_{i \in C_k} \theta_{i0} X_i + \sum_{i \in C_k, j \in MB_k, k' \in MBC_k} \theta_{ij} X_i \langle X_j \rangle_{q(X_{C_k})} \right\}$$

Virtually a reparameterized Ising model of small size.
GMF approximation to Ising models

Attractive coupling: positively weighted
Repulsive coupling: negatively weighted

© Eric Xing @ CMU, 2005-2015
Example 2: Sigmoid belief network
Example 3: Factorial HMM
Currently for each new model we have to
- derive the variational update equations
- write application-specific code to find the solution

Each can be time consuming and error prone

Can we build a general-purpose inference engine which automates these procedures?
Cluster-based MF (e.g., GMF)

- a general, iterative message passing algorithm
- clustering completely defines approximation
  - preserves dependencies
  - flexible performance/cost trade-off
  - clustering automatable
- recovers model-specific structured VI algorithms, including:
  - fHMM, LDA
  - variational Bayesian learning algorithms
- easily provides new structured VI approximations to complex models