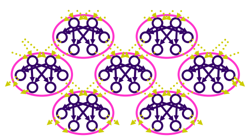
Advanced Machine Learning

Variational Inference

Eric Xing

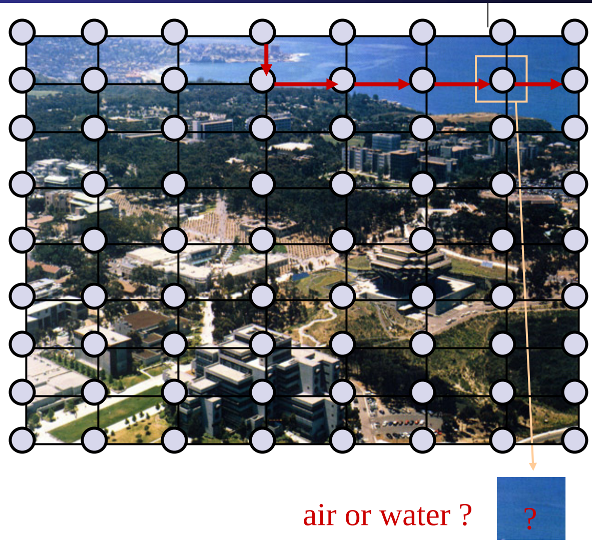
Lecture 12, August 12, 2009



Reading:

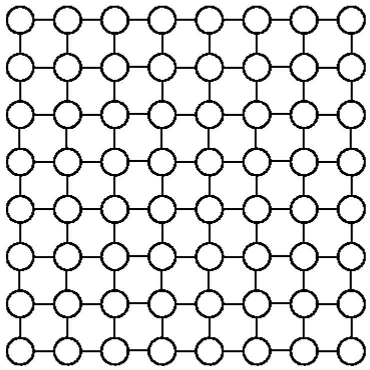
An Ising model on 2-D image

- 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.



Why Approximate Inference?

- Tree-width of NxN graph is O(N)
- N can be a huge number(~1000s of pixels)
- Exact inference will be too expensive



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

Variational Methods

- 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:

$$\text{e.g. } f^* = \arg\max_{f \in S} \text{ or } \min \left\{ F(f) \right\}$$

f : a (tractable) probability distribution
or, solutions to certain probabilistic queries

Bethe Energy Minimization

The Objective

- 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 to P
- 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 $Q_1=Q_2$
  - But, $KL(Q_1||Q_2) \neq KL(Q_2||Q_1)$

Which KL?

- Computing KL(P||Q) requires inference!
- But KL(P||Q) 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 $P(X)=1/Z\prod_{f_a\in F} f_a(X_a)$

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

The Objective

-

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

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

*also called Gibbs Free Energy

The Energy Functional

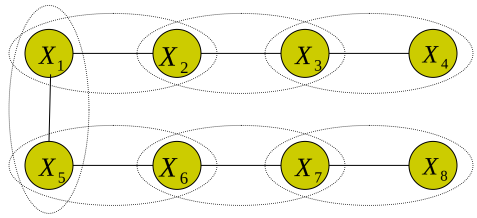
- Let us look at the functional

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

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

Tree Energy Functionals

- Consider a tree-structured distribution



- The probability can be written as: $b(\mathbf{x}) = \prod_{i,j \in E} b_{ij}(x_i, x_j) \prod_i b_i(x_i)^{-1}$
- $H_{tree} = -\sum_{i,j \in E} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \ln b_{ij}(x_i, x_j) + \sum_i 1 \times \sum_{x_i} b_i(x_i) \ln b_i(x_i)$
- $F_{Tree} = -\left(-\sum_{i,j \in E} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \ln b_{ij}(x_i, x_j) + \sum_i 1 \times \sum_{x_i} b_i(x_i) \ln b_i(x_i) \right) - \sum_{i,j \in E} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \ln f_{i,j}(x_i, x_j) - \sum_i \sum_{x_i} b_i(x_i) \ln f_i(x_i)$

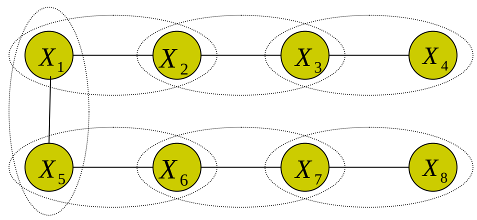
$$= \sum_{i,j \in E} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \ln \frac{b_{ij}(x_i, x_j)}{f_{i,j}(x_i, x_j)} + \sum_i \sum_{x_i} b_i(x_i) \ln \frac{b_i(x_i)}{f_i(x_i)} - \sum_i 2 \times \sum_{x_i} b_i(x_i) \ln b_i(x_i)$$

$$= F_{12} + F_{23} + .. + F_{67} + F_{78} - F_1 - F_5 - F_2 - F_6 - F_3 - F_7$$

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

Tree Energy Functionals

- Consider a tree-structured distribution



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

  $= F_{12} + F_{23} + .. + F_{67} + F_{78} - F_1 - F_5 - F_2 - F_6 - F_3 - F_7$

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

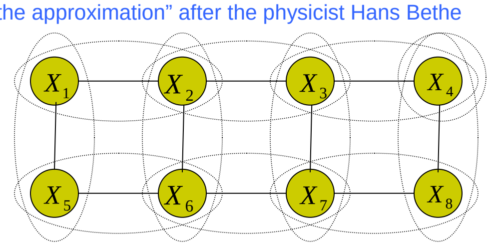
Bethe Approximation to Gibbs Free Energy

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

  $H_{Bethe} = -\sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln b_a(\mathbf{x}_a) + \sum_i (d_i - 1) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i)$

  $F_{Bethe} = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln \frac{b_a(\mathbf{x}_a)}{f_a(\mathbf{x}_a)} + \sum_i (1 - d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i) = -\langle f_a(\mathbf{x}_a) \rangle - H_{betha}$

  - Called "Bethe approximation" after the physicist Hans Bethe



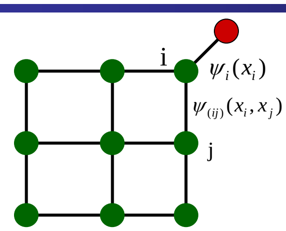
$$F_{bethe} = F_{12} + F_{23} + .. + F_{67} + F_{78} - F_1 - F_5 - 2F_2 - 2F_6 .. - F_8$$

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

Bethe Approximation

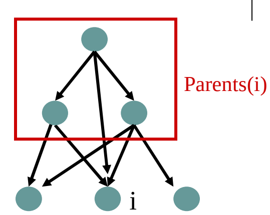
- Pros:
  - Easy to compute, since entropy term involves sum over pairwise and single variables
- Cons:
  - $\hat{F}(P,Q) = F_{bethe}$ **may or may not** be well connected to $F(P,Q)$
  - It could, in general, be greater, equal or less than $F(P,Q)$
- Optimize each $b(\mathbf{x}_a)$'s.
  - For discrete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

From GM to factored graphs



Undirected graph
(Markov random field)

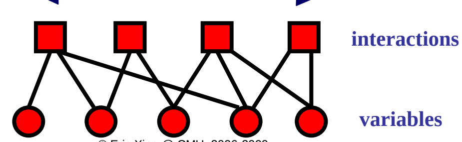
$$P(x) = \frac{1}{Z}\prod_i \psi_i(x_i) \prod_{(ij)} \psi_{(ij)}(x_i, x_j)$$



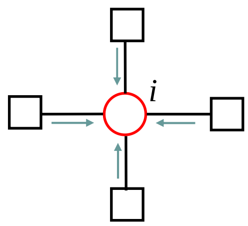
Directed graph
(Bayesian network)

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

factor graphs

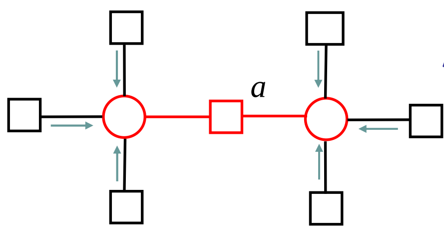


Recall Beliefs and messages in FG



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

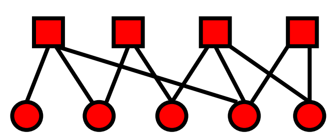
"beliefs" "messages"



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

The "belief" is the BP approximation of the marginal probability.

Bethe Free Energy for FG



$$F_{Betha} = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln \frac{b_a(\mathbf{x}_a)}{f_a(\mathbf{x}_a)} + \sum_i (1-d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i)$$

$$H_{Bethe} = -\sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln b_a(\mathbf{x}_a) + \sum_i (d_i - 1) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i)$$

$$F_{Bethe} = -\langle f_a(\mathbf{x}_a) \rangle - H_{betha}$$

Constrained Minimization of the Bethe Free Energy

$$L = F_{Bethe} + \sum_i \gamma_i \{\sum_{x_i} b_i(x_i) - 1\}$$

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

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

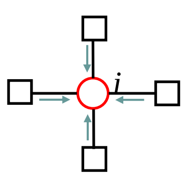
$$\frac{\partial L}{\partial b_a(X_a)} = 0 \quad \Longrightarrow \quad b_a(X_a) \propto \exp\left(-E_a(X_a) + \sum_{i \in N(a)} \lambda_{ai}(x_i) \right)$$

Bethe = BP on FG

- Identify

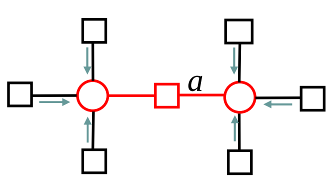
$$\lambda_{ai}(x_i) = \ln \prod_{b \in N(i) \neq a} m_{b \to i}(x_i)$$

- to obtain BP equations:



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

"beliefs" "messages"



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

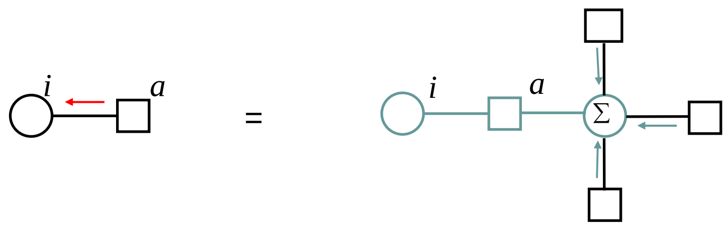
The "belief" is the BP approximation of the marginal probability.

BP Message-update Rules

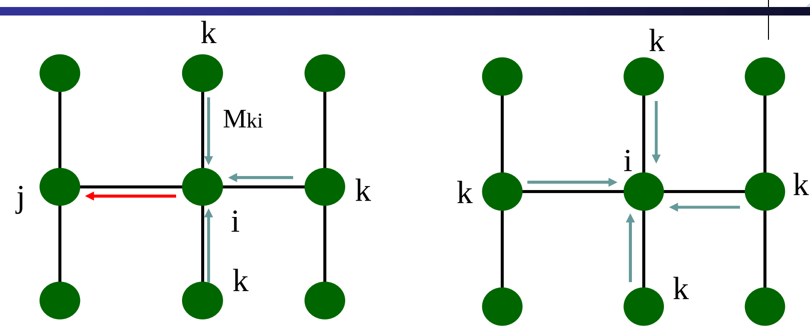
Using $b_{a\to i}(x_i) = \sum_{X_a \backslash x_i} b_a(X_a)$, we get

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

(A sum product algorithm)



Belief Propagation on trees



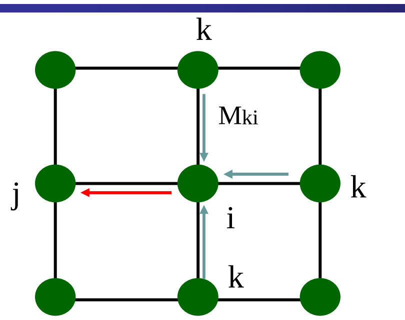
- BP Message-update Rules

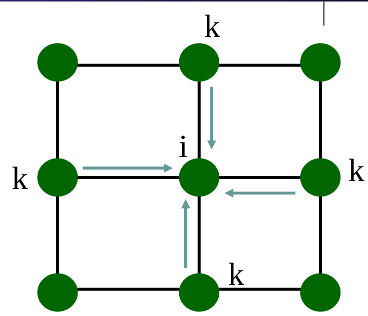
$$M_{i\to j}(x_j) \propto \sum_{x_i} \psi_{ij}(x_i, x_j)\psi_i(x_i)\prod_k M_{k\to i}(x_i) \qquad b_i(x_i) \propto \psi_i(x_i)\prod_k M_k(x_k)$$

Compatibilities (interactions) external evidence

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

Belief Propagation on loopy graphs





- BP Message-update Rules

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

external evidence

Compatibilities (interactions)

$$b_i(x_i) \propto \psi_i(x_i)\prod_k M_k(x_k)$$

- May not converge or converge to a wrong solution

Loopy Belief Propagation

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

The Theory Behind LBP

- For a distribution $p(\mathrm{X}|\theta)$ 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:

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

$$F_{Bethe} = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln \frac{b_a(\mathbf{x}_a)}{f_a(\mathbf{x}_a)} + \sum_i (1-d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i) = -\langle f_a(\mathbf{x}_a) \rangle - H_{bethe}$$

q : a (tractable) probability distribution

The Theory Behind LBP

- But we do not optimize $q(\mathbf{X})$ 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: $H_{Betha} = H(b_{i,j}, b_i)$
  - relaxed feasible set: $\mathcal{M}_o = \left\{ \tau \geq 0 \mid \sum_{x_i} \tau(x_i) = 1, \sum_{x_i} \tau(x_i, x_j) = \tau(x_j) \right\}$

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

- The loopy B
  - a fixed point iteration procedure that tries to solve b^*

Mean Field Approximation

Mean field methods

- Optimize $q(\mathbf{X}_H)$ 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: $\mathcal{Q} \to \mathcal{T} \quad (\mathcal{T} \subseteq \mathcal{Q})$

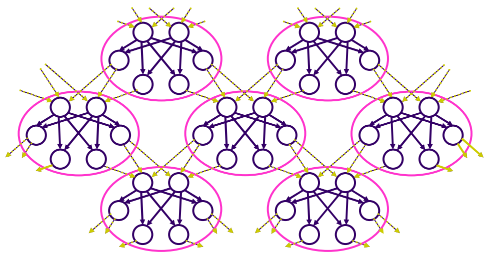
$$q^* = \arg\min_{q \in \mathcal{T}} \langle E \rangle_q - H_q$$

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, $\{C_1, \dots, C_I\}$, of all variables
- Let $$q(\mathbf{X}) = \prod_i q_i(\mathbf{X}_{C_i}),$$
- Mean-field free energy

$$G_{\mathrm{MF}} = \sum_i \sum_{\mathbf{x}_{C_i}} \prod_i q_i(\mathbf{x}_{C_i}) E(\mathbf{x}_{C_i}) + \sum_i \sum_{\mathbf{x}_{C_i}} q_i(\mathbf{x}_{C_i}) \ln q_i(\mathbf{x}_{C_i})$$

e.g., $$G_{\mathrm{MF}} = \sum_{i<j} \sum_{x_i x_j} q(x_i) q(x_j) \phi(x_i x_j) + \sum_i \sum_{x_i} q(x_i) \phi(x_i) + \sum_i \sum_{x_i} q(x_i) \ln q(x_i)$$ (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 $q_i(x_c)$'s.
  - Variational calculus …
  - Do inference in each $q_i(x_c)$ using any tractable algorithm

The Generalized Mean Field theorem

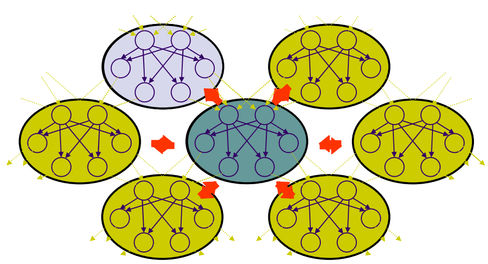
Theorem: The optimum GMF approximation to the cluster marginal is isomorphic to the cluster posterior of the original distribution given internal evidence and its generalized mean fields:

$$q_i^*(\mathbf{X}_{H,C_i}) = p(\mathbf{X}_{H,C_i} \mid \mathbf{x}_{E,C_i}, \langle \mathbf{X}_{H,MB_i} \rangle_{q_{j \neq i}})$$

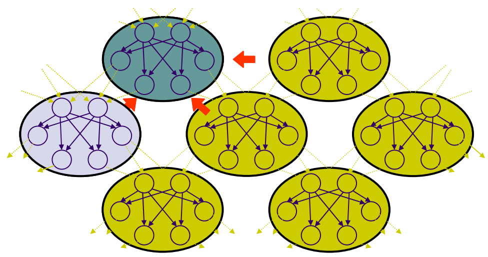
GMF algorithm: Iterate over each q_i

A generalized mean field algorithm

[xing *et al*. UAI 2003]



A generalized mean field algorithm [xing *et al*. UAI 2003]



Convergence theorem

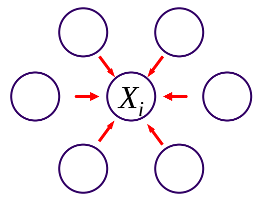
Theorem: The GMF algorithm is guaranteed to converge to a local optimum, and provides a lower bound for the likelihood of evidence (or partition function) the model.

The naive mean field approximation

- Approximate $p(\mathbf{X})$ by fully factorized $q(\mathbf{X})=\prod_i q_i(X_i)$
- For Boltzmann distribution $p(X)=\exp\{\sum_{i<j} \theta_{ij}X_iX_j+\theta_{io}X_i\}/Z$:

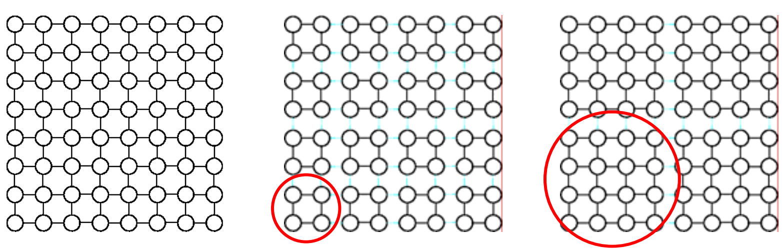
mean field equation:

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



- $\langle X_j\rangle_{q_j}$ resembles a "message" sent from node j to i
- $\{\langle X_j\rangle_{q_j} : j\in\mathcal{N}_i\}$ forms the "mean field" applied to X_i from its neighborhood

Generalized MF approximation to Ising models

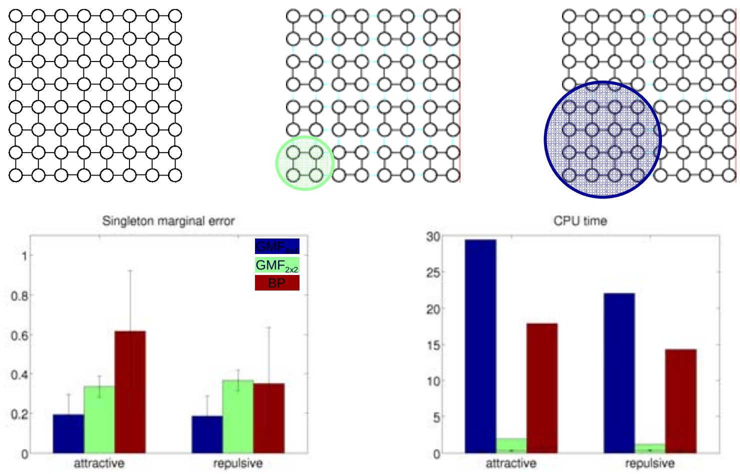


Cluster marginal of a square block C_k:

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

Virtually a reparameterized Ising model of small size.

GMF approximation to Ising models



Attractive coupling: positively weighted
Repulsive coupling: negatively weighted

Automatic Variational Inference

fHMM **Mean field approx.** **Structured variational approx.**

- 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

Example: Factorial HMM