Probabilistic Graphical Models

Lecture 11: Variational Inference (Continued)

Matthew Walter

TTI-Chicago

May 15, 2020

Approximate Inference (Revisited)

- Given a graphical model over $\boldsymbol{X}$ and evidence $\boldsymbol{E} = \boldsymbol{e}$, we are interested in conditional probability queries $P(\boldsymbol{Y} \mid \boldsymbol{E} = \boldsymbol{e})$ for $\boldsymbol{Y} \subseteq \boldsymbol{X}$
- While exact inference is NP-hard, several real-world inference problems are easy (e.g., hidden Markov models)
- However, exact inference is intractable for many problems
- *Approximate inference* provides a tractable alternative
- Nearly all approximate algorithms are either:
  1. Variational algorithms (e.g., mean field, loopy belief propagation)
  2. Monte-carlo methods (e.g., MCMC)
- This and the previous lecture focus on variational methods

Variational Methods (Revisited)

- **Goal**: Approximate a difficult distribution $P(\boldsymbol{X} \mid \boldsymbol{e})$ with a new distribution $Q(\boldsymbol{X})$ such that:
  1. $P(\boldsymbol{X} \mid \boldsymbol{e})$ and $Q(\boldsymbol{X})$ are "close"
  2. Inference on $Q(\boldsymbol{X})$ is easy
- How should we measure the distance between distributions?
- The **Kullback-Liebler divergence** (KL-divergence) between two distributions P and Q is defined as

$$D(P \| Q) = \mathbb{E}_P \left[\log \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})} \right] \qquad D(Q \| P) = \mathbb{E}_Q \left[\log \frac{Q(\boldsymbol{x})}{P(\boldsymbol{x})} \right]$$

- $D(P \| Q) \geq 0 \; \forall \; P$ and Q and zero iff $P = Q$ (similarly for $D(Q \| P)$)
- KL-divergence is **not symmetric**, i.e., $D(P \| Q) \neq D(Q \| P)$

KL-Divergence (Revisited)

$$D(P\|Q) = \mathbb{E}_P \left[\log \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})} \right] \qquad D(Q\|P) = \mathbb{E}_Q \left[\log \frac{Q(\boldsymbol{x})}{P(\boldsymbol{x})} \right]$$

- Let P be the true distribution that we want to perform inference over
- **M-projection**:

$$Q^*_M = \arg\min_Q D(P\|Q)$$

- **I-projection**:

$$Q^*_I = \arg\min_Q D(Q\|P)$$

- These two will differ when Q is minimized over a restricted set of distributions, i.e., $\mathcal{Q} = \{Q_1, \ldots, Q_n\}$, where $P \notin \mathcal{Q}$
- Solving for Q^*_M is as difficult as exact inference over P

Variational Methods (Revisited)

$$
\begin{aligned}
D(Q\|P) &= -\left\{ \sum_{c\in\mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X})) \right\} + \ln Z(\theta) \\
&= -F[\tilde{P}, Q] + \ln Z(\theta)
\end{aligned}
$$

where $\theta_c(\boldsymbol{X}_c) = \ln \phi(\boldsymbol{X}_c)$

- $F[\tilde{P}, Q]$ is the (negative) **variational (Helmholtz) free energy**
  - The first *energy term* $\sum_{c\in\mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] = \mathbb{E}_Q\left[\sum_{c\in\mathcal{C}} \ln \phi_c(\boldsymbol{X}_c)\right]$ involves expectations over (bounded) factors
  - The second *entropy term* is the entropy over Q
- The complexity of computing both terms is a function of Q (not P)
- We can force Q to be closer to P by maximizing the energy functional

Variational Methods: Optimizing the Energy Functional

$$\max_Q \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- What is the space of distributions $\mathcal{Q}$ that we are optimizing over?
  - Define an "easy" family of distributions $\mathcal{Q}$
  - Assume a factorized form that offers convenient structure
- The objective function is concave in Q, **but** there are exponentially many distributions $Q(x)$
- Two general approaches:
  1. Optimize the *exact* energy functional, but restricted to a space of (simpler) distributions (that generally do not include P)
  2. Optimize an *approximate* energy functional

Variational Methods: Optimizing the Energy Functional

$$\max_{Q} \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- What is the space of distributions $\mathcal{Q}$ that we are optimizing over?
  - Define an "easy" family of distributions $\mathcal{Q}$
  - Assume a factorized form that offers convenient structure
- The objective function is concave in Q, **but** there are exponentially many distributions $Q(x)$
- **Relaxation** algorithms (last lecture) operate directly on *pseudomarginals* that may not be consistent with any joint distribution (approximate energy functional)
- **Structured variational** algorithms (today) optimize the exact energy functional over a family $\mathcal{Q}$ of tractable, *coherent* distributions

Structured Variational Inference

- Approach: Choose $\mathcal{Q}$ with enough *structure* to afford efficient inference while providing a good approximation of P_Φ

Structured Variational Inference

- Approach: Choose $\mathcal{Q}$ with enough *structure* to afford efficient inference while providing a good approximation of P_Φ
- Letting $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_M\}$ be some partition of $\boldsymbol{X}$, restrict $\mathcal{Q}$ to the class of distributions that factorize according to these partitions

$$Q(\boldsymbol{X}) = \prod_{i}^{M} Q(\boldsymbol{X}_i)$$

Structured Variational Inference

- Approach: Choose $\mathcal{Q}$ with enough *structure* to afford efficient inference while providing a good approximation of P_Φ
- Letting $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_M\}$ be some partition of $\boldsymbol{X}$, restrict $\mathcal{Q}$ to the class of distributions that factorize according to these partitions

$$Q(\boldsymbol{X}) = \prod_i^M Q(\boldsymbol{X}_i)$$

- In the **Naive Mean Field** model, partitions are individual variables

$$Q(\boldsymbol{X}) = \prod_i Q(X_i)$$

Structured Variational Inference

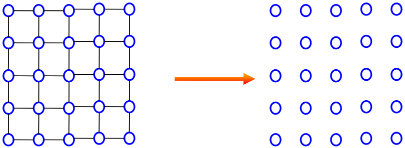
- Approach: Choose $\mathcal{Q}$ with enough *structure* to afford efficient inference while providing a good approximation of P_Φ
- Letting $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_M\}$ be some partition of $\boldsymbol{X}$, restrict $\mathcal{Q}$ to the class of distributions that factorize according to these partitions

$$Q(\boldsymbol{X}) = \prod_i^M Q(\boldsymbol{X}_i)$$

- In the **Naive Mean Field** model, partitions are individual variables

$$Q(\boldsymbol{X}) = \prod_i Q(X_i)$$

Example: Pairwise Markov networks (e.g., classify image pixels)



Naive Mean Field

$$Q(\boldsymbol{X}) = \prod_i Q(X_i)$$

- We can use this to simplify the optimization of the energy functional

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- Note that $Q(\boldsymbol{X}_c) = \prod_{i \in c} Q(X_i)$
- Note that the joint entropy decomposes as a sum of local entropies:

$$H(Q(\boldsymbol{X})) = -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln Q(\boldsymbol{X})$$

Naive Mean Field

$$Q(\boldsymbol{X}) = \prod_i Q(X_i)$$

- We can use this to simplify the optimization of the energy functional

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- Note that $Q(\boldsymbol{X}_c) = \prod_{i \in c} Q(X_i)$
- Note that the joint entropy decomposes as a sum of local entropies:

$$\begin{aligned} H(Q(\boldsymbol{X})) &= -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln Q(\boldsymbol{X}) \\ &= -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln \prod_i Q(X_i) = -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \sum_i \ln Q(X_i) \end{aligned}$$

Naive Mean Field

$$Q(\boldsymbol{X}) = \prod_i Q(X_i)$$

- We can use this to simplify the optimization of the energy functional

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- Note that $Q(\boldsymbol{X}_c) = \prod_{i \in c} Q(X_i)$
- Note that the joint entropy decomposes as a sum of local entropies:

$$\begin{aligned} H(Q(\boldsymbol{X})) &= -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln Q(\boldsymbol{X}) \\ &= -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln \prod_i Q(X_i) = -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \sum_i \ln Q(X_i) \\ &= -\sum_i \sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln Q(X_i) \end{aligned}$$

Naive Mean Field

$$Q(\boldsymbol{X}) = \prod_i Q(X_i)$$

- We can use this to simplify the optimization of the energy functional

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \mathbb{E}_Q[\theta_c(\boldsymbol{X}_c)] + H(Q(\boldsymbol{X}))$$

- Note that $Q(\boldsymbol{X}_c) = \prod_{i \in c} Q(X_i)$
- Note that the joint entropy decomposes as a sum of local entropies:

$$\begin{aligned}
H(Q(\boldsymbol{X})) &= -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln Q(\boldsymbol{X}) \\
&= -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln \prod_i Q(X_i) = -\sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \sum_i \ln Q(X_i) \\
&= -\sum_i \sum_{\boldsymbol{X}} Q(\boldsymbol{X}) \ln Q(X_i) \\
&= -\sum_i \sum_{X_i} Q(X_i) \ln Q(X_i) \sum_{\boldsymbol{X}-\{i\}} Q(\boldsymbol{X}_{\boldsymbol{X}-\{i\}} \mid X_i) = \sum_i H(Q(X_i))
\end{aligned}$$

Naive Mean Field

- Putting these together, we get the following variational objective

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}_c} \theta_c(\boldsymbol{X}_c) \prod_{i \in c} Q(X_i) + \sum_i H(Q(X_i))$$

subject to the constraints

$$\begin{aligned} Q(x_i) &\geq 0 \qquad && \forall i, x_i \in Val(X_i) \\ \sum_{x_i \in Val(X_i)} Q(x_i) &= 1 \qquad && \forall i \end{aligned}$$

Naive Mean Field

- Putting these together, we get the following variational objective

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}_c} \theta_c(\boldsymbol{X}_c) \prod_{i \in c} Q(X_i) + \sum_i H(Q(X_i))$$

  subject to the constraints

$$\begin{aligned} Q(x_i) &\geq 0 \qquad && \forall i, x_i \in Val(X_i) \\ \sum_{x_i \in Val(X_i)} Q(x_i) &= 1 \qquad && \forall i \end{aligned}$$

- Unlike relaxation methods, which optimize an approximate objective over pseudomarginals, mean field optimizes the true objective and approximates the optimization space $\boldsymbol{Q}$

Naive Mean Field

- Putting these together, we get the following variational objective

$$\max_{Q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}_c} \theta_c(\boldsymbol{X}_c) \prod_{i \in c} Q(X_i) + \sum_i H(Q(X_i))$$

subject to the constraints

$$\begin{aligned} Q(x_i) &\geq 0 \qquad && \forall i, x_i \in Val(X_i) \\ \sum_{x_i \in Val(X_i)} Q(x_i) &= 1 \qquad && \forall i \end{aligned}$$

- Distribution $Q(X_i)$ is a *local maximum* given $\{Q_j(X_j)\}_{j \neq i}$ iff

$$Q(X_i) = \frac{1}{Z_i} \exp\left(\sum_{c \in \mathcal{C}} \overbrace{\sum_{\boldsymbol{X}_c} Q(\boldsymbol{X}_c \mid X_i) \theta_c(\boldsymbol{X}_c)}^{\mathbb{E}_Q[\theta_c(\boldsymbol{X}_c) \mid X_i]} \right)$$

where Z_i is a local normalizing constant

Naive Mean Field

- The Lagrangian associated with this optimization over each $Q(X_i)$ is

$$L_i[Q] = \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}_c} \theta_c(\boldsymbol{X}_c) \prod_{i \in c} Q(X_i) + \sum_i H(Q(X_i)) + \lambda \left(\sum_{x_i} Q(x_i) - 1 \right)$$

Naive Mean Field

- The Lagrangian associated with this optimization over each $Q(X_i)$ is

$$L_i[Q] = \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}_c} \theta_c(\boldsymbol{X}_c) \prod_{i \in c} Q(X_i) + \sum_i H(Q(X_i)) + \lambda \left(\sum_{x_i} Q(x_i) - 1 \right)$$

- Taking the partial derivative with respect to $Q(x_i)$ yields

$$\frac{\partial}{\partial Q(x_i)} L_i = \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}_c} \theta_c(\boldsymbol{X}_c) Q(\boldsymbol{X}_c \mid x_i) - \ln Q(x_i) - 1 + \lambda$$

Naive Mean Field

- The Lagrangian associated with this optimization over each $Q(X_i)$ is

$$L_i[Q] = \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}_c} \theta_c(\boldsymbol{X}_c) \prod_{i \in c} Q(X_i) + \sum_i H(Q(X_i)) + \lambda \left(\sum_{x_i} Q(x_i) - 1 \right)$$

- Taking the partial derivative with respect to $Q(x_i)$ yields

$$\frac{\partial}{\partial Q(x_i)} L_i = \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}_c} \theta_c(\boldsymbol{X}_c) Q(\boldsymbol{X}_c \mid x_i) - \ln Q(x_i) - 1 + \lambda$$

- Setting this to zero and rearranging terms, we get

$$\ln Q(x_i) = \lambda - 1 + \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}_c} \theta_c(\boldsymbol{X}_c) Q(\boldsymbol{X}_c \mid x_i)$$

Naive Mean Field (Continued)

- Setting this to zero and rearranging terms, we get

$$\ln Q(x_i) = \lambda - 1 + \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{X}_c} \theta_c(\boldsymbol{X}_c) Q(\boldsymbol{X}_c \,|\, x_i)$$

- Taking the exponent and renormalizing (λ is a constant), and since the objective is concave in $Q(X_i)$ given all other elements of Q, we get the following theorem

Theorem: Distribution $Q(X_i)$ is a *local maximum* (fixed point) given $\{Q_j(X_j)\}_{j \neq i}$ iff

$$Q(x_i) = \frac{1}{Z_i} \exp\left(\sum_{c \in \mathcal{C}} \overbrace{\sum_{\boldsymbol{X}_c} Q(\boldsymbol{X}_c \,|\, x_i) \theta_c(\boldsymbol{X}_c)}^{\mathbb{E}_Q[\theta_c(\boldsymbol{X}_c) \,|\, x_i]} \right)$$

where Z_i is a local normalizing constant

Naive Mean Field (Continued)

- We have the following expression for the fixed point

$$Q(x_i) = \frac{1}{Z_i} \exp \left(\sum_{c \in \mathcal{C}} \overbrace{\sum_{\boldsymbol{X}_c} Q(\boldsymbol{X}_c \,|\, x_i) \theta_c(\boldsymbol{X}_c)}^{\mathbb{E}_Q[\theta_c(\boldsymbol{X}_c) \,|\, x_i]} \right)$$

- Since $Q(\boldsymbol{X}_c \,|\, x_i) = Q(\boldsymbol{X}_c)$ is independent of X_i, we can move these terms into the normalization constant Z_i

$$Q(x_i) = \frac{1}{Z_i} \exp \left(\sum_{c : X_i \in Scope(c)} \overbrace{\sum_{\boldsymbol{X}_c} Q(\boldsymbol{X}_c \,|\, x_i) \theta_c(\boldsymbol{X}_c)}^{\mathbb{E}_Q[\theta_c(\boldsymbol{X}_c) \,|\, x_i]} \right)$$

- $Q(X_i)$ only has to be consistent with the expectation of the (log) potentials θ in which it appears

Naive Mean Field for Pairwise MRFs

- Consider a pairwise MRF (e.g., foreground/background estimation)

$$\max_{Q \in \mathcal{Q}} \sum_{i,j \in E} \sum_{x_i, x_j} \theta_{i,j}(x_i, x_j) Q(x_i) Q_j(x_j) - \sum_{i} \sum_{x_i \in Val(X_i)} Q(x_i) \ln Q(x_i)$$

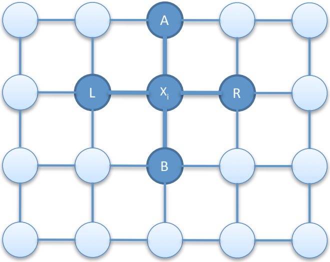
- The expression for the fixed point of each $Q(X_i)$ given all other elements of Q is

$$Q(x_i) \leftarrow \frac{1}{Z_i} \exp \left(\theta_i(x_i) + \sum_{j \in N(i)} \sum_{x_j \in Val(X_j)} Q_j(x_j) \theta_{i,j}(x_i, x_j) \right)$$

Naive Mean Field for Pairwise MRFs

$$Q(x_i) \leftarrow \frac{1}{Z_i} \exp\left(\theta_i(x_i) + \sum_{j \in N(i)} \sum_{x_j \in Val(X_j)} Q_j(x_j)\theta_{i,j}(x_i, x_j)\right)$$

$$\begin{aligned}
\phi_i(\text{fg}) &= \exp \frac{-\|c_i - \mu_{\text{fg}}\|^2}{\sigma^2} \\
\phi_i(\text{bg}) &= \exp \frac{-\|c_i - \mu_{\text{bg}}\|^2}{\sigma^2} \\
\phi_{i,j}(X_i, X_j) &= \begin{cases} 10 & \text{if } X_i = X_j \\ 1 & \text{otherwise} \end{cases}
\end{aligned}$$



$$Q_{X_i}(\text{fg}) = \frac{1}{Z_i} \exp \begin{pmatrix} \log \phi_i(\text{fg}) & + & & \\ Q_A(\text{fg}) \log \phi_{A,X_i}(\text{fg}, \text{fg}) & + & Q_A(\text{bg}) \log \phi_{A,X_i}(\text{bg}, \text{fg}) & + \\ Q_B(\text{fg}) \log \phi_{B,X_i}(\text{fg}, \text{bg}) & + & Q_B(\text{bg}) \log \phi_{B,X_i}(\text{bg}, \text{fg}) & + \\ Q_L(\text{fg}) \log \phi_{L,X_i}(\text{fg}, \text{fg}) & + & Q_L(\text{bg}) \log \phi_{L,X_i}(\text{bg}, \text{fg}) & + \\ Q_R(\text{fg}) \log \phi_{R,X_i}(\text{fg}, \text{fg}) & + & Q_R(\text{bg}) \log \phi_{R,X_i}(\text{bg}, \text{fg}) & \end{pmatrix}$$

Naive Mean Field for Pairwise MRFs

$$Q(x_i) \leftarrow \frac{1}{Z_i} \exp\left(\theta_i(x_i) + \sum_{j \in N(i)} \sum_{x_j \in Val(X_j)} Q_j(x_j)\theta_{i,j}(x_i, x_j)\right)$$

- This is a non-convex optimization problem with many local maxima!
- We can greedily optimize it using block coordinate ascent
  1. For each $i \in V$
     - Fully maximize above equation w.r.t. $\{Q(x_i) \ \ \forall x_i \in Val(X_i)\}$
  2. Repeat until convergence

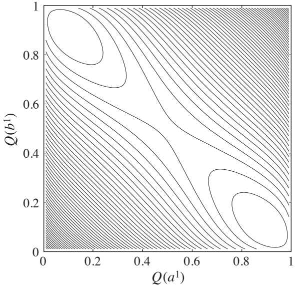
$$Q_{X_i}(\mathrm{fg}) \quad = \quad \frac{1}{Z_i} \exp \left(\begin{array}{rcrc} \log \phi_i(\mathrm{fg}) & + & & \\ Q_A(\mathrm{fg}) \log \phi_{A,X_i}(\mathrm{fg},\mathrm{fg}) & + & Q_A(\mathrm{bg}) \log \phi_{A,X_i}(\mathrm{bg},\mathrm{fg}) & + \\ Q_B(\mathrm{fg}) \log \phi_{B,X_i}(\mathrm{fg},\mathrm{bg}) & + & Q_B(\mathrm{bg}) \log \phi_{B,X_i}(\mathrm{bg},\mathrm{fg}) & + \\ Q_L(\mathrm{fg}) \log \phi_{L,X_i}(\mathrm{fg},\mathrm{fg}) & + & Q_L(\mathrm{bg}) \log \phi_{L,X_i}(\mathrm{bg},\mathrm{fg}) & + \\ Q_R(\mathrm{fg}) \log \phi_{R,X_i}(\mathrm{fg},\mathrm{fg}) & + & Q_R(\mathrm{bg}) \log \phi_{R,X_i}(\mathrm{bg},\mathrm{fg}) & \end{array} \right)$$

Mean Field Convergence

- With coordinate ascent, every step of the mean field algorithm increases the energy functional
- Each mean field iteration yields a better approximation Q of the target distribution P_Φ
- Mean field algorithm is guaranteed to converge
- At convergence, we have a stationary point
  - Could be a local minimum, local maximum, or a saddle point
  - In practice, it is usually a local maximum
- We can use multiple random restarts to avoid local maxima
- However, the approximation fundamentally can not capture complex distributions

Mean Field Convergence: Example

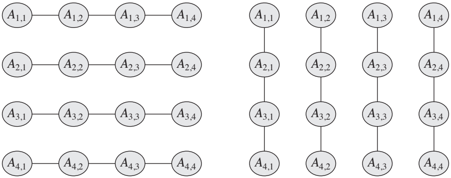
- Consider a distribution that represents an approximate XOR of A and B: $P(a, b) = 0.5 - \epsilon$ if $a \neq b$ and $P(a, b) = \epsilon$ if $a = b$
- Can not accurately approximate P by a product of marginals
- When ϵ is small, the energy functional has two local maxima corresponding to $a \neq b$



Level sets of the energy functional

- When $\epsilon > 0.1$, mean field approximation has a single maximum

Structured Approximations



Two possible factorizations for a 4×4 Ising model

- It is often useful to consider factorizations over partitions $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_M\}$ that include more than one random variable

$$Q(\boldsymbol{X}) = \prod_{i}^{M} Q(\boldsymbol{X}_i)$$

- This allows us to capture relationships present in P_Φ and, in turn, better approximate P_Φ
- However, we need to balance the improvements to the approximation with the cost of inference using Q

Software Packages

1. libDAI
   - http://www.libdai.org
   - Implements several exact and approximate inference methods: Exact inference via junction-trees, mean field, loopy belief propagation, ...
2. Infer.NET
   - http://research.microsoft.com/en-us/um/cambridge/projects/infernet/
   - Provides implementations of several machine learning algorithms
   - Includes implementations of mean field and loopy sum-product belief propagation
   - Handles continuous variables