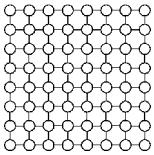
Machine Learning

10-701/15-781, Fall 2011

Undirected Graphical Models
And
Approximate Inference

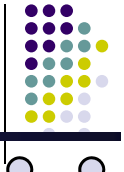


Eric Xing

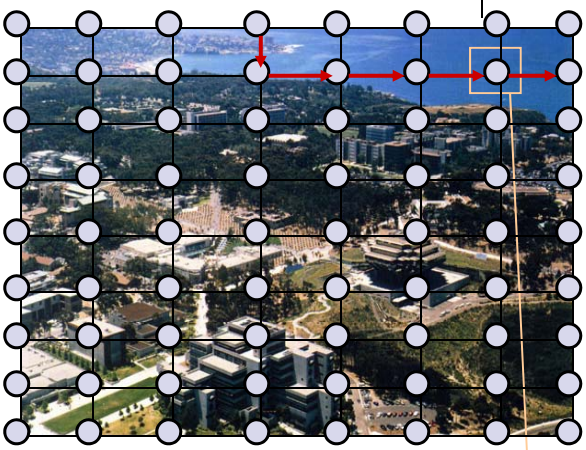
Lecture 15, November 2, 2011

Reading: Chap. 8, C.B book

What is in the image?

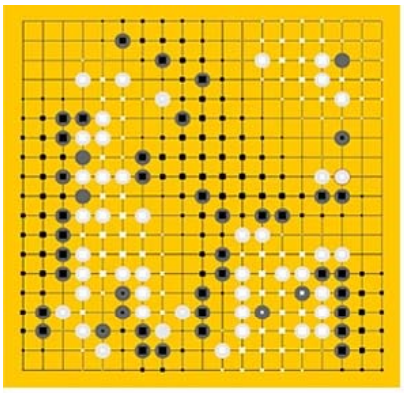


- Nodes encode hidden information (patch-identity).
- They receive local information from the image (brightness, color).
- Information is propagated though the graph over its edges.
- Edges encode 'compatibility' between nodes.



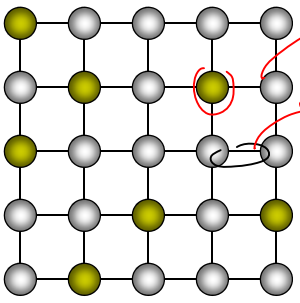
air or water ?

Modeling Go



This is the middle position of a Go game. Overlaid is the estimate for the probability of becoming black or white for every intersection. Large squares mean the probability is higher.

Ising Model



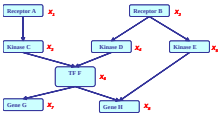
$$p(X) = \frac{1}{Z}\exp\left\{\sum_{i<j}\theta_{ij}X_iX_j + \sum_i \theta_{i0}X_i\right\}$$

- Naturally arises in image processing, lattice physics, etc.
- Each node may represent a single "pixel", or an atom
  - The states of adjacent or nearby nodes are "coupled" due to pattern continuity or electro-magnetic force, etc.
  - Most likely joint-configurations usually correspond to a "low-energy" state

Two types of GMs

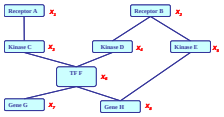
- **Directed edges give causality relationships (Bayesian Network or Directed Graphical Model):**

$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$
$$= P(X_1)\,P(X_2)\,P(X_3|X_1)\,P(X_4|X_2)\,P(X_5|X_2)\,P(X_6|X_3, X_4)\,P(X_7|X_6)\,P(X_8|X_5, X_6)$$



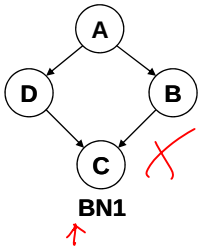
- **Undirected edges simply give correlations between variables (Markov Random Field or Undirected Graphical model):**

$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$
$$= \frac{1}{Z}\exp\{E(X_1)+E(X_2)+E(X_3, X_1)+E(X_4, X_2)+E(X_5, X_2)+E(X_6, X_3, X_4)+E(X_7, X_6)+E(X_8, X_5, X_6)\}$$

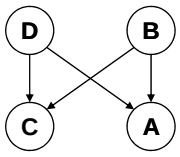


P-maps

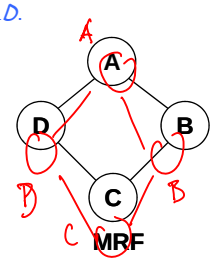
- Defn: A DAG G is a **perfect map** (P-map) for a distribution P if $I(P)=I(G)$.
- Thm: not every distribution has a perfect map as DAG.
  - Pf by counterexample. Suppose we have a model where $A \perp C \mid \{B,D\}$, and $B \perp D \mid \{A,C\}$. This cannot be represented by any Bayes net.
  - e.g., BN1 wrongly says $B \perp D \mid A$, BN2 wrongly says $B \perp D$.



BN1

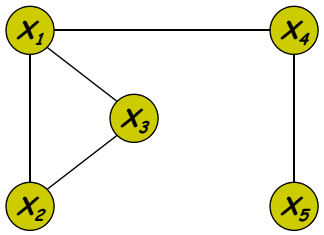


BN2



MRF

Undirected graphical models (UGM)



- Pairwise (non-causal) relationships
- Can write down model, and score specific configurations of the graph, but no explicit way to generate samples
- Contingency constrains on node configurations

Representation

- Defn: an undirected graphical model represents a distribution $P(X_1,\ldots,X_n)$ defined by an undirected graph H, and a set of positive ***potential functions*** y_c associated with cliques of H, s.t.

$$P(x_1,\ldots,x_n) = \frac{1}{Z}\prod_{c\in C}\psi_c(\mathbf{x}_c)$$

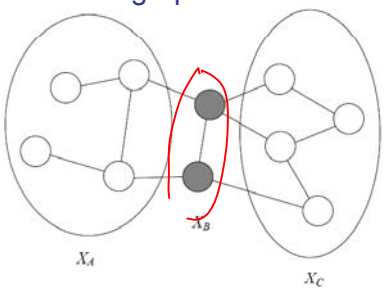
where Z is known as the partition function:

$$Z = \sum_{x_1,\ldots,x_n}\prod_{c\in C}\psi_c(\mathbf{x}_c)$$

- Also known as Markov Random Fields, Markov networks …
- The ***potential function*** can be understood as an contingency function of its arguments assigning "pre-probabilistic" score of their joint configuration.

Global Markov Independencies

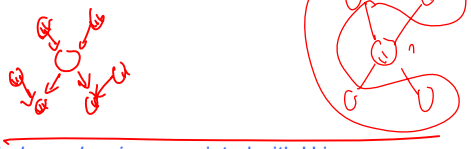
- Let *H* be an undirected graph:



- *B* ***separates*** *A* and *C* if every path from a node in *A* to a node in *C* passes through a node in *B*: $\text{sep}_H(A;C|B)$
- A probability distribution satisfies the ***global Markov property*** if for any disjoint *A*, *B*, *C*, such that *B* separates *A* and *C*, *A* is independent of *C* given *B:* $\text{I}(H) = \{A \perp C | B : \text{sep}_H(A;C|B)\}$

Local Markov independencies

- For each node $X_i \in \mathbf{V}$, there is *unique Markov blanket* of X_i, denoted MB_{Xi}, which is the set of neighbors of X_i in the graph (those that share an edge with X_i)



- **Defn:**

The *local Markov independencies* associated with H is:

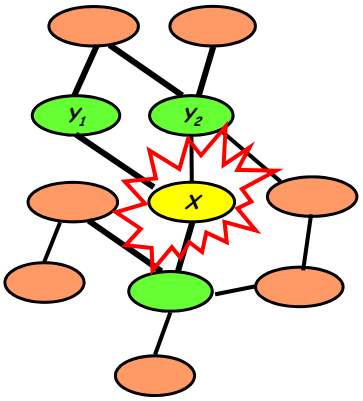
$$I_\ell(H): \{X_i \perp \mathbf{V} - \{X_i\} - MB_{Xi} \mid MB_{Xi} : \forall\ i\},$$

In other words, X_i is independent of the rest of the nodes in the graph given its immediate neighbors

Summary: Conditional Independence Semantics in an MRF

Structure: an *undirected graph*

- **Meaning: a node is conditionally independent of every other node in the network given its Directed neighbors**
- **Local contingency functions (potentials) and the cliques in the graph completely determine the joint dist.**
- **Give correlations between variables, but no explicit way to generate samples**

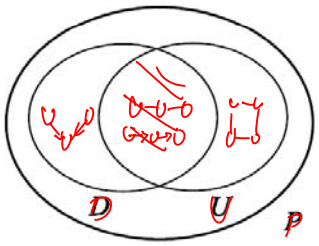


Perfect maps

- Defn: A Markov network H is a perfect map for P if for any X; Y;Z we have that

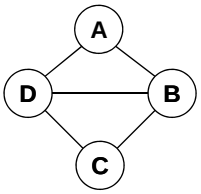
$$\text{sep}_H(X;Z|Y) \Leftrightarrow P \models (X \perp Z \mid Y)$$

- Thm: not every distribution has a perfect map as UGM.
  - Pf by counterexample. No undirected network can capture all and only the independencies encoded in a v-structure $X \rightarrow Z \leftarrow Y$.



Quantitative Specification: Cliques

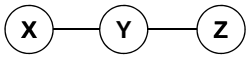
- For $G=\{V,E\}$, a complete subgraph (clique) is a subgraph $G'=\{V'\subseteq V, E'\subseteq E\}$ such that nodes in V' are fully interconnected
- A (maximal) clique is a complete subgraph s.t. any **superset** $V''\supset V'$ is not complete.
- A sub-clique is a not-necessarily-maximal clique.



$P(\ ) = \frac{1}{Z}\prod_c \phi(X_c)$

- Example:
  - max-cliques = {A,B,D}, {B,C,D},
  - sub-cliques = {A,B}, {C,D}, …→ all edges and singletons

Interpretation of Clique Potentials



- The model implies $X \perp Z \mid Y$. This independence statement implies (by definition) that the joint must factorize as:

$$p(x,y,z) = p(y)p(x\mid y)p(z\mid y)$$

- We can write this as: $p(x,y,z) = p(x,y)p(z\mid y)$, but $p(x,y,z) = p(x\mid y)p(z,y)$
  - **cannot** have all potentials be marginals
  - **cannot** have all potentials be conditionals

- The positive clique potentials can only be thought of as general "compatibility", "goodness" or "happiness" functions over their variables, but not as probability distributions.

Hammersley-Clifford Theorem

- If arbitrary potentials are utilized in the following product formula for probabilities,

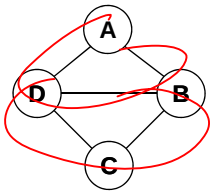
$$P(x_1,\ldots,x_n) = \frac{1}{Z}\prod_{c\in C}\psi_c(\mathbf{x}_c)$$

$$Z = \sum_{x_1,\ldots,x_n}\prod_{c\in C}\psi_c(\mathbf{x}_c)$$

then the family of probability distributions obtained is exactly that set which **respects** the *qualitative specification* (the conditional independence relations) described earlier

- **Thm :** Let P be a positive distribution over $\mathbf{V}$, and *H* a Markov network graph over $\mathbf{V}$. If *H* is an I-map for P, then P is a Gibbs distribution over *H*.

Example:



$$P(x_1,x_2,x_3,x_4)$$
$$= \frac{1}{Z}\psi_c(\mathbf{x}_{124})\times\psi_c(\mathbf{x}_{234})$$
$$\times\psi_{12}(\mathbf{x}_{12})\psi_{14}(\mathbf{x}_{14})\psi_{23}(\mathbf{x}_{23})\psi_{24}(\mathbf{x}_{24})\psi_{34}(\mathbf{x}_{34})$$
$$\times\psi_1(x_1)\psi_2(x_2)\psi_3(x_3)\psi_4(x_4)$$

$$Z = \sum_{x_1,x_2,x_3,x_4} \psi_c(\mathbf{x}_{124})\times\psi_c(\mathbf{x}_{234})$$
$$\times\psi_{12}(\mathbf{x}_{12})\psi_{14}(\mathbf{x}_{14})\psi_{23}(\mathbf{x}_{23})\psi_{24}(\mathbf{x}_{24})\psi_{34}(\mathbf{x}_{34})$$
$$\times\psi_1(x_1)\psi_2(x_2)\psi_3(x_3)\psi_4(x_4)$$

Exponential Form

- Constraining clique potentials to be positive could be inconvenient (e.g., the interactions between a pair of atoms can be either attractive or repulsive). We represent a clique potential $\psi_c(\mathbf{x}_c)$ in an unconstrained form using a real-value "energy" function $\phi_c(\mathbf{x}_c)$:

$$\psi_c(\mathbf{x}_c) = \exp\{-\phi_c(\mathbf{x}_c)\}$$

  For convenience, we will call $\phi_c(\mathbf{x}_c)$ a potential when no confusion arises from the context.

- This gives the joint a nice additive strcuture

$$p(\mathbf{x}) = \frac{1}{Z}\exp\left\{-\sum_{c\in C}\phi_c(\mathbf{x}_c)\right\} = \frac{1}{Z}\exp\{-H(\mathbf{x})\}$$

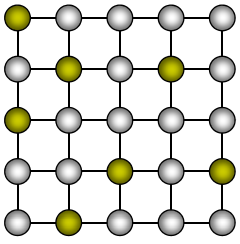
  where the sum in the exponent is called the "free energy":

$$H(\mathbf{x}) = \sum_{c\in C}\phi_c(\mathbf{x}_c)$$

- In physics, this is called the "Boltzmann distribution".
- In statistics, this is called a log-linear model.

Example: Ising models

- Nodes are arranged in a regular topology (often a regular packing grid) and connected only to their geometric neighbors.



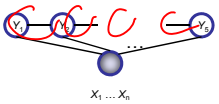
$$p(X) = \frac{1}{Z}\exp\left\{\sum_{i,j\in N_i}\theta_{ij}X_iX_j + \sum_i \theta_{i0}X_i\right\}$$

- Same as sparse Boltzmann machine, where $\theta_{ij}\neq 0$ iff i,j are neighbors.
  - e.g., nodes are pixels, potential function encourages nearby pixels to have similar intensities.
- Potts model: multi-state Ising model.

Example: Conditional Random Fields

- If the graph $G = (V, E)$ of $\mathbf{Y}$ is a tree, the conditional distribution over the label sequence $\mathbf{Y} = \mathbf{y}$, given $\mathbf{X} = \mathbf{x}$, by the Hammersley Clifford theorem of random fields is:

$$p_\theta(\mathrm{y} \mid \mathrm{x}) \propto \exp\left(\sum_{e \in E,k} \lambda_k f_k(e, \mathrm{y}|_e, \mathrm{x}) + \sum_{v \in V,k} \mu_k g_k(v, \mathrm{y}|_v, \mathrm{x}) \right)$$



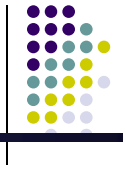
  - x is a data sequence
  - y is a label sequence
  - *v* is a vertex from vertex set V = set of label random variables
  - *e* is an edge from edge set E over V
  - f_k and g_k are given and fixed. g_k is a Boolean vertex feature; f_k is a Boolean edge feature
  - *k* is the number of features
  - $\theta = (\lambda_1, \lambda_2, \cdots, \lambda_n; \mu_1, \mu_2, \cdots, \mu_n)$; λ_k and μ_k are parameters to be estimated
  - $\mathrm{y}|_e$ is the set of components of y defined by edge *e*
  - $\mathrm{y}|_v$ is the set of components of y defined by vertex *v*

Inference and Learning

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

- Learning:
  - Parameter estimation
  - Structure estimation

MLE for undirected graphical models

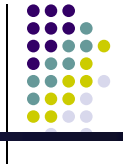


- For directed graphical models, the log-likelihood decomposes into a sum of terms, one per family (node plus parents).
- For undirected graphical models, the log-likelihood does not decompose, because the normalization constant Z is a function of **all** the parameters

$$P(x_1,\ldots,x_n)=\frac{1}{Z}\prod_{c\in C}\psi_c(\mathbf{x}_c) \qquad Z=\sum_{x_1,\ldots,x_n}\prod_{c\in C}\psi_c(\mathbf{x}_c)$$

- In general, we will need to do inference (i.e., marginalization) to learn parameters for undirected models, even in the fully observed case.

Inference Problems



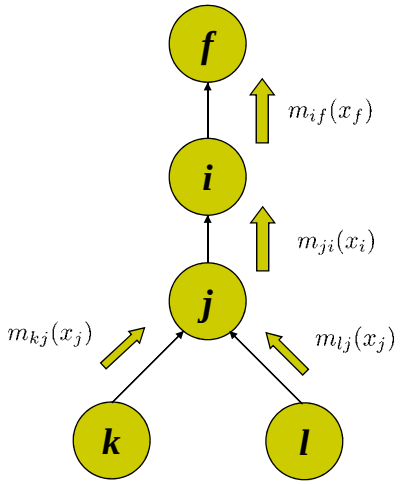
- Compute the likelihood of observed data
- Compute the marginal distribution $p(x_A)$ over a particular subset of nodes $A \subset V$
- Compute the conditional distribution $p(x_A|x_B)$ for disjoint subsets *A* and *B*
- Compute a mode of the density $\hat{x}=\arg\max_{x\in\mathcal{X}^m} p(x)$

- Methods we have

Brute force | Elimination → **Message Passing** (Forward-backward , Max-product /BP, Junction Tree)

Individual computations independent — Sharing intermediate terms

Message passing for trees



Let $m_{ij}(x_i)$ denote the factor resulting from eliminating variables from bellow up to i, which is a function of x_i:

$$m_{ji}(x_i) = \sum_{x_j} \left(\psi(x_j)\psi(x_i, x_j) \prod_{k \in N(j) \setminus i} m_{kj}(x_j) \right)$$

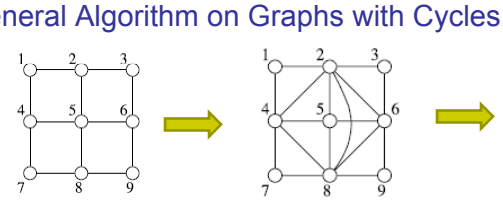
This is reminiscent of a *message* sent from j to i.

$$p(x_f) \propto \psi(x_f) \prod_{e \in N(f)} m_{ef}(x_f)$$

$m_{ij}(x_i)$ represents a "belief" of x_i from x_j!

Junction Tree Revisited

- General Algorithm on Graphs with Cycles



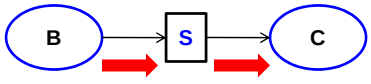
- Steps:

=> **Triangularization** => **Construct JTs**

=> Message Passing on Clique Trees

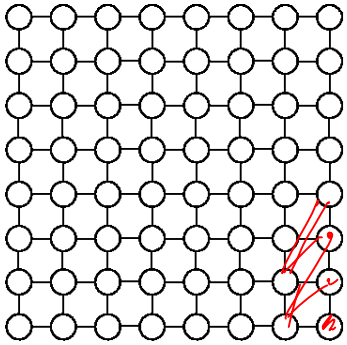
$$\widetilde{\phi}_S(x_S) \leftarrow \sum_{x_{B \setminus S}} \phi_B(x_B)$$

$$\phi_C(x_C) \leftarrow \frac{\widetilde{\phi}_S(x_S)}{\phi_S(x_S)} \phi_C(x_C)$$



Why Approximate Inference?

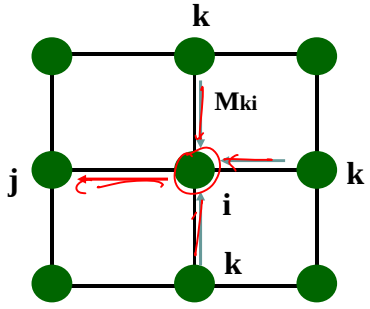
- Why can't we just run junction tree on this graph?

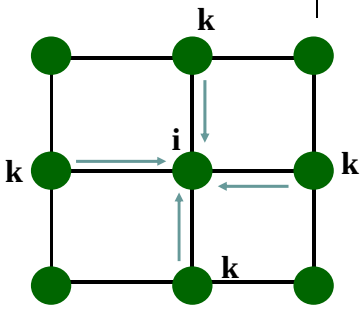


$$p(X) = \frac{1}{Z}\exp\left\{\sum_{i<j}\theta_{ij}X_iX_j + \sum_i \theta_{i0}X_i\right\}$$

- If NxN 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

Solution 1: Belief Propagation on loopy graphs





- BP Message-update Rules

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

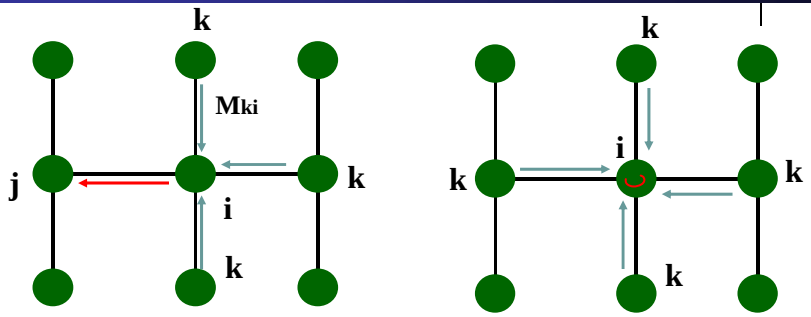
Compatibilities (interactions)

external evidence

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

- May not converge or converge to a wrong solution

Recall BP 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

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

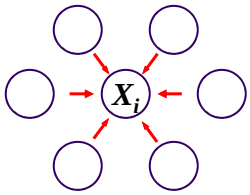
Solution 2: The naive mean field approximation

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

mean field equation:

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

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



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

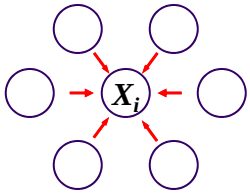
Recall Gibbs sampling

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

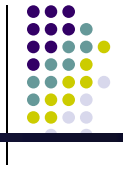
Gibbs predictive distribution:

$$p(X_i \mid x_{-i}) = \exp\left\{\theta_{i0}X_i + \sum_{j\in\mathcal{N}_i}\theta_{ij}X_i x_j + A_i\right\}$$

$$= p(X_i \mid \{x_j : j\in\mathcal{N}_i\})$$



Supplemental reading



Theoretical Foundation of Approx Inference

- 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

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

  - $KL(Q_1||Q_2)>=0$
  - $KL(Q_1||Q_2)=0$ iff $Q_1=Q_2$
  - But, $KL(Q_1||Q_2) \neq KL(Q_2||Q_1)$

The Objective

- $$KL(Q \| P) = \underbrace{-H_Q(X) - \sum_{f_a \in F} E_Q \log f_a(X_a)}_{F(P,Q)} + \log Z$$
- We will call $F(P,Q)$ the "Energy Functional" , or, the Gibbs Free Energy
- $F(P,P)$ =?
- $F(P,Q)$ >= $F(P,P)$

The Energy Functional

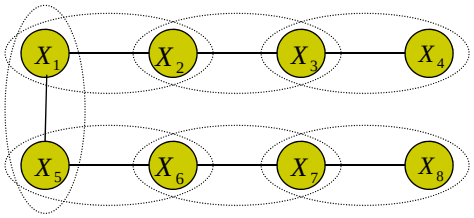
- Let us look at the functional

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

- $\sum_{f_a \in F} E_Q \log f_a(X_a)$ 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: 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: $b(\mathbf{x}) = \prod_a b_a(\mathbf{x}_a) \prod_i b_i(x_i)^{1-d_i}$
- $$H_{tree} = -\sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln b_a(\mathbf{x}_a) + \sum_i (d_i - 1) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i)$$
- $$F_{Tree} = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln \frac{b_a(\mathbf{x}_a)}{f_a(\mathbf{x}_a)} + \sum_i (1 - d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i)$$
$$= F_{12} + F_{23} + .. + F_{67} + F_{78} - F_1 - F_5 - F_2 - F_6 - F_3 - F_7$$
  - involves summation over edges and vertices and is therefore easy to compute

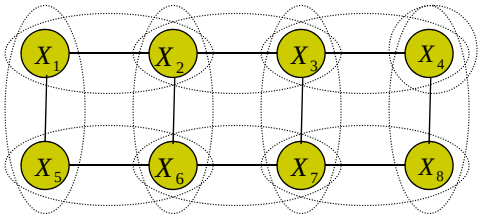
Bethe Approximation to Gibbs Free Energy

- 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

Constrained Minimization of the Bethe Free Energy

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

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

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

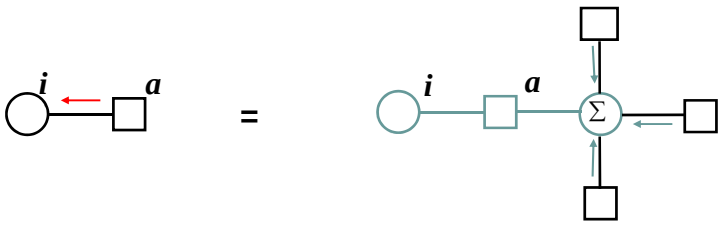
$$\frac{\partial L}{\partial b_a(X_a)} = 0 \quad \Rightarrow \quad b_a(X_a) \propto \exp\left(-E_a(X_a) + \sum_{i \in N(a)} \lambda_{ai}(x_i) \right)$$

Bethe = BP Message-update Rules

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

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

(A sum product algorithm)



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: Approximate $F(P,Q)$ with easy to compute $\hat{F}(P,Q)$

Mean field approx. to Gibbs free energy

- Given a disjoint clustering, $\{C_1, \ldots, C_J\}$, of all variables
- Let
$$q(\mathbf{X}) = \prod_i q_i(\mathbf{X}_{C_i}),$$
- Mean-field free energy
$$G_{\text{MF}} = \sum_i \sum_{\mathbf{x}_{C_i}} \prod_i q_i(\mathbf{x}_{C_i}) E(\mathbf{x}_{C_i}) + \sum_i \sum_{\mathbf{x}_{C_i}} q_i(\mathbf{x}_{C_i}) \ln q_i(\mathbf{x}_{C_i})$$

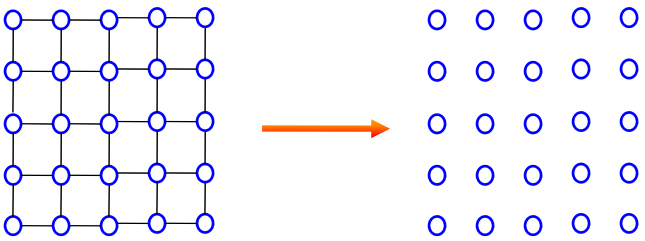
e.g., $G_{\text{MF}} = \sum_{i<j} \sum_{x_i x_j} q(x_i) q(x_j) \phi(x_i x_j) + \sum_i \sum_{x_i} q(x_i) \phi(x_i) + \sum_i \sum_{x_i} q(x_i) \ln q(x_i)$ **(naïve mean field)**

  - 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

Naïve Mean Field

- Fully factorized variational distribution

$$q(x) = \prod_{s \in V} q(x_s)$$



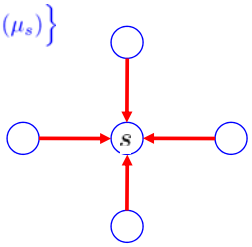
Naïve Mean Field for Ising Model

- 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

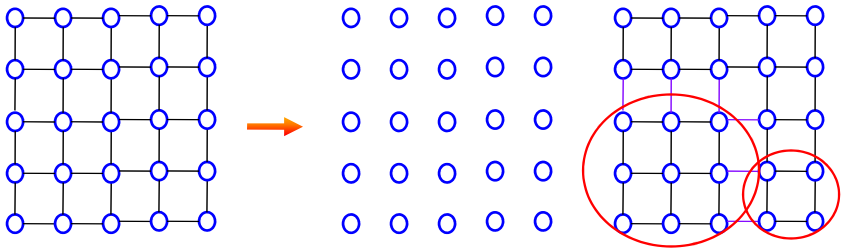
$$\mu_s \leftarrow \sigma\left(\theta_s + \sum_{t \in N(s)} \theta_{st} \mu_t\right)$$



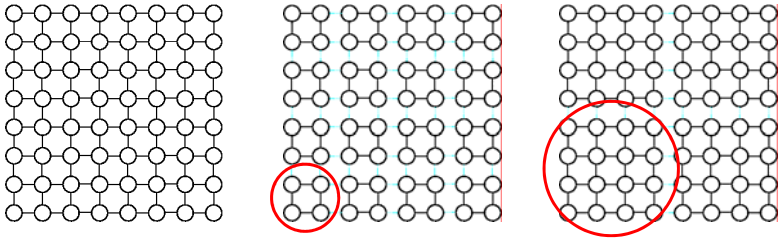
  - $\mu_t = p(X_t = 1) = \mathbb{E}_p[X_t]$ resembles "message" sent from node t to s
  - $\{\mathbb{E}_p[X_t], t \in N(s)\}$ forms the "mean field" applied to s from its neighborhood

Structured Mean Field

- Mean field theory is general to any tractable sub-graphs
- Naïve mean field is based on the fully unconnected sub-graph
- Variants based on structured sub-graphs can be derived



Generalized MF approximation to Ising models [Xing et al. 2003]



Cluster marginal of a square block C_k:

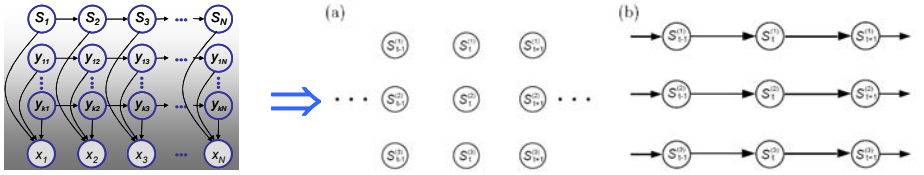
$$q(X_{C_k}) \propto \exp\left\{ \sum_{i,j \in C_k} \theta_{ij} X_i X_j + \sum_{i \in C_k} \theta_{i0} X_i + \sum_{\substack{i \in C_k, j \in MB_k, \\ k' \in MBC_k}} \theta_{ij} X_i \left\langle X_j \right\rangle_{q(X_{C_{k'}})} \right\}$$

Virtually a reparameterized Ising model of small size.

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

Structured Variational Inference



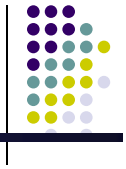
fHMM Mean field approx. Structured variational approx.

- 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?

Summary I

- Undirected graphical models capture "relatedness", "coupling", "co-occurrence", "synergism", etc. between entities
- Local and global independence properties identifiable via graph separation criteria
- Defined on clique potentials
- Generally intractable to compute likelihood due to presence of "partition function"
  - Therefore not only inference, but also likelihood-based learning is difficult in general
- Can be used to define either joint or conditional distributions
- Important special cases:
  - Gaussian graphical models
  - Ising models

Summary II



- Exact inference methods are limited to tree-structured graphs
- Junction Tree methods is exponentially expensive to the tree-width
- Message Passing methods can be applied for loopy graphs, but lack of analysis!
- Mean-field is convergent, but can have local optimal
- **Where do these two algorithm come from? Do they make sense?**