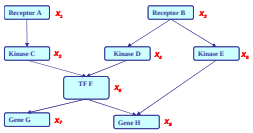
School of Computer Science
Carnegie Mellon

Graphical Models (II)

Inference



Eric Xing

Carnegie Mellon University
June 1, 2007

Eric Xing,
A lecture series at the Institute of Theoretical Computer Science, Tsinghua University, May 31-June 7, 2007

Recap of Basic Prob. Concepts

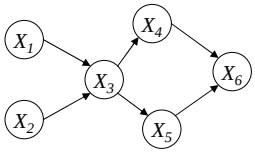
- Joint probability dist. on multiple variables:

$$P(X_1, X_2, X_3, X_4, X_5, X_6)$$
$$= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1, X_2)P(X_4 \mid X_1, X_2, X_3)P(X_5 \mid X_1, X_2, X_3, X_4)P(X_6 \mid X_1, X_2, X_3, X_4, X_5)$$

- If X_i's are **independent**: ($P(X_i|\cdot)= P(X_i)$)

$$P(X_1, X_2, X_3, X_4, X_5, X_6)$$
$$= P(X_1)P(X_2)P(X_3)P(X_4)P(X_5)P(X_6) = \prod_i P(X_i)$$

- If X_i's are **conditionally independent** (as described by a **GM**), the joint can be factored to simpler products, e.g.,



$$P(X_1, X_2, X_3, X_4, X_5, X_6)$$
$$= P(X_1)P(X_2)P(X_3 \mid X_1, X_2)P(X_4 \mid X_3)P(X_5 \mid X_3)P(X_6 \mid X_4, X_5)$$

Inference and Learning

- We now have compact representations of probability distributions: **Graphical Models**
- A GM *M* describes a unique probability distribution *P*
- Typical tasks:
  - Task 1: How do we answer **queries** about P?
    - We use **inference** as a name for the process of computing answers to such queries
  - Task 2: How do we estimate a **plausible model** M from data D?
    i. We use **learning** as a name for the process of obtaining point estimate of M.
    ii. But for *Bayesian*, they seek $p(M \mid D)$, which is actually an **inference** problem.
    iii. When not all variables are observable, even computing point estimate of M need to do **inference** to impute the *missing data*.

Inferential Query 1: Likelihood

- Most of the queries one may ask involve **evidence**
  - Evidence $\mathbf{x}_v$ is an assignment of values to a set $\mathbf{X}_v$ of nodes in the GM over varialbe set $\mathbf{X}=\{X_1, X_2, \ldots, X_n\}$
  - Without loss of generality $\mathbf{X}_v=\{X_{k+1}, \ldots, X_n\}$,
  - Write $\mathbf{X}_H=\mathbf{X}\backslash\mathbf{X}_v$ as the set of hidden variables, $\mathbf{X}_H$ can be $\emptyset$ or $\mathbf{X}$
- Simplest query: compute probability of evidence

$$P(\mathbf{x}_v) = \sum_{\mathbf{x}_H} P(\mathbf{X}_H, \mathbf{X}_v) = \sum_{x_1} \cdots \sum_{x_k} P(x_1, \ldots, x_k, \mathbf{x}_v)$$

  - this is often referred to as computing the **likelihood** of $\mathbf{x}_v$

Inferential Query 2: Conditional Probability

- Often we are interested in the **conditional probability distribution** of a variable given the evidence

$$P(\mathbf{X}_H \mid \mathbf{X}_V = \mathbf{x}_V) = \frac{P(\mathbf{X}_H, \mathbf{x}_V)}{P(\mathbf{x}_V)} = \frac{P(\mathbf{X}_H, \mathbf{x}_V)}{\sum_{\mathbf{x}_H} P(\mathbf{X}_H = \mathbf{x}_H, \mathbf{x}_V)}$$

  - this is the ***a posteriori* belief** in $\mathbf{X}_H$, given evidence $\mathbf{x}_v$

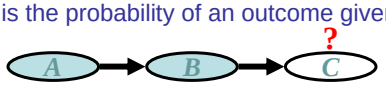
- We usually query a subset $\mathbf{Y}$ of all hidden variables $\mathbf{X}_H$=**{Y,Z}** and "don't care" about the remaining, $\mathbf{Z}$:

$$P(\mathbf{Y} \mid \mathbf{x}_V) = \sum_{\mathbf{z}} P(\mathbf{Y}, \mathbf{Z} = \mathbf{z} \mid \mathbf{x}_V)$$

  - the process of summing out the "don't care" variables z is called **marginalization**, and the resulting $P(\mathbf{Y}|\mathbf{x}_v)$ is called a **marginal** prob.

Applications of *a posteriori* Belief

- **Prediction**: what is the probability of an outcome given the starting condition



  - the query node is a descendent of the evidence

- **Diagnosis**: what is the probability of disease/fault given symptoms

  - the query node an ancestor of the evidence

- **Learning** under partial observation
  - fill in the unobserved values under an "EM" setting (more later)

- The directionality of information flow between variables is not restricted by the directionality of the edges in a GM
  - probabilistic inference can combine evidence form all parts of the network

Inferential Query 3: Most Probable Assignment

- In this query we want to find the **most probable joint assignment** (MPA) for ***some*** variables of interest
- Such reasoning is usually performed under some given evidence $\mathbf{x}_V$, and ignoring (the values of) other variables $\mathbf{Z}$:

$$\mathbf{Y}^* \mid \mathbf{x}_V = \arg\max_y P(\mathbf{Y} \mid \mathbf{x}_V) = \arg\max_y \sum_{\mathbf{z}} P(\mathbf{Y}, \mathbf{Z} = \mathbf{z} \mid \mathbf{x}_V)$$

  - this is the **maximum *a posteriori*** configuration of $\mathbf{Y}$.

Applications of MPA

- Classification
  - find most likely label, given the evidence
- Explanation
  - what is the most likely scenario, given the evidence

Cautionary note:

- The MPA of a variable depends on its "context"---the set of variables been jointly queried
- Example:
  - MPA of X ?
  - MPA of (X, Y) ?

x	y	$P(x,y)$
0	0	0.35
0	1	0.05
1	0	0.3
1	1	0.3

Complexity of Inference

Thm:

Computing $P(\mathbf{X}_H=\mathbf{x}_H|\ \mathbf{x}_v)$ in an arbitrary GM is NP-hard

- Hardness does not mean we cannot solve inference
  - It implies that we cannot find a general procedure that works efficiently for arbitrary GMs
  - For particular families of GMs, we can have provably efficient procedures

Approaches to inference

- Exact inference algorithms
  - The sum-product algorithm
  - The junction tree algorithm √
- Approximate inference techniques
  - Stochastic simulation / sampling methods
  - Markov chain Monte Carlo methods
  - Variational algorithms (later lectures) √

The Junction Tree Algorithm

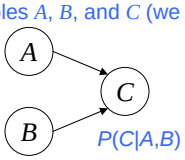
- There are several inference algorithms; some of which operate directly on (special) directed graph
  - Forward-backward algorithm for HMM (we will see it later)
  - Pealing algorithm for trees and phylogenies
- The junction tree algorithm is the most popular and general inference algorithm, it operates on an undirected graph
  - To understand the JT-algorithm, we need to understand how to compile a directed graph into an undirected graph

Moral Graph

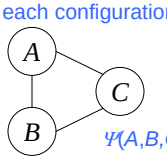
- Note that for both directed GMs and undirected GMs, the joint probability is in a product form:

BN: $P(\mathbf{X}) = \prod_{i=1:d} P(X_i \mid \mathbf{X}_{\pi_i})$ MRF: $P(\mathbf{X}) = \frac{1}{Z}\prod_{c \in C} \psi_c(\mathbf{X}_c)$

- So let's convert local conditional probabilities into potentials; then the second expression will be generic, but how does this operation affect the directed graph?
  - We can think of a conditional probability, e.g,. $P(C|A,B)$ as a function of the three variables A, B, and C (we get a real number of each configuration):



$P(C|A,B)$

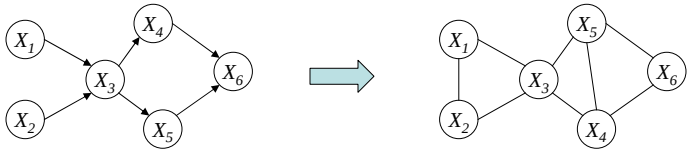


$\Psi(A,B,C) = P(C|A,B)$

  - Problem: But a node and its parent are not generally in the same clique in a BN
  - Solution: Marry the parents to obtain the "moral graph"

Moral Graph (cont.)

- Define the potential on a clique as the product over all conditional probabilities contained within the clique
- Now the product of potentials gives the right answer:



$$P(X_1, X_2, X_3, X_4, X_5, X_6)$$
$$= P(X_1)P(X_2)P(X_3 \mid X_1, X_2)P(X_4 \mid X_3)P(X_5 \mid X_3)P(X_6 \mid X_4, X_5)$$
$$= \psi(X_1, X_2, X_3)\psi(X_3, X_4, X_5)\psi(X_4, X_5, X_6)$$

where $\psi(X_1, X_2, X_3) = P(X_1)P(X_2)P(X_3 \mid X_1, X_2)$

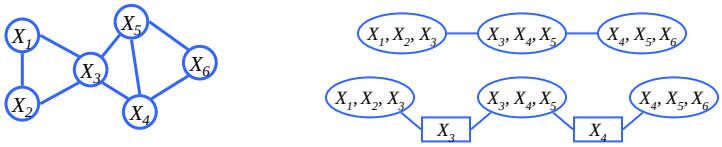
$\psi(X_3, X_4, X_5) = P(X_4 \mid X_3)P(X_5 \mid X_3)$

$\psi(X_4, X_5, X_6) = P(X_6 \mid X_4, X_5)$

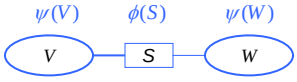
Note that here the interpretation of potential is ambivalent:
it can be either *marginals* or *conditionals*

Clique trees

- A clique tree is an (undirected) tree of cliques



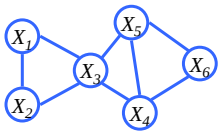
- Consider cases in which two neighboring cliques V and W have an overlap S (e.g., (X_1, X_2, X_3) overlaps with (X_3, X_4, X_5)),

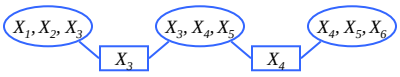


- Now we have an alternative representation of the joint in terms of the potentials:

Clique trees

- A clique tree is an (undirected) tree of cliques





- The alternative representation of the joint in terms of the potentials:

$$
\begin{aligned}
& P(X_1, X_2, X_3, X_4, X_5, X_6) \\
&= P(X_1)P(X_2)P(X_3 \mid X_1, X_2)P(X_4 \mid X_3)P(X_5 \mid X_3)P(X_6 \mid X_4, X_5) \\
&= P(X_1, X_2, X_3)\frac{P(X_3, X_4, X_5)}{P(X_3)}\frac{P(X_4, X_5, X_6)}{P(X_4, X_5)} \\
&= \psi(X_1, X_2, X_3)\frac{\psi(X_3, X_4, X_5)}{\phi(X_3)}\frac{\psi(X_4, X_5, X_6)}{\phi(X_4, X_5)}
\end{aligned}
$$

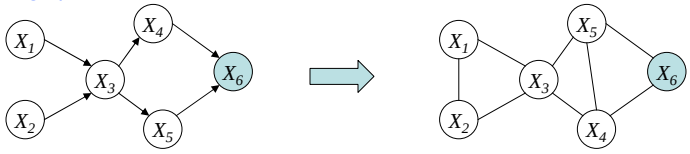
Now each potential is isomorphic to the *cluster marginal* of the attendant set of variables

- Generally:

$$P(\mathbf{X}) = \frac{\prod_C \psi_C(\mathbf{X}_C)}{\prod_S \phi_S(\mathbf{X}_S)}$$

Why this is useful?

- Propagation of probabilities
  - Now suppose that some evidence has been "absorbed" (i.e., certain values of some nodes have been observed). How do we propagate this effect to the rest of the graph?



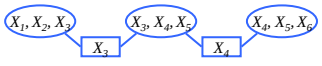
  - What do we mean by propagate?

    Can we adjust all the potentials $\{\psi\}$, $\{\phi\}$ so that they still represent the correct cluster marginals (or unnormalized equivalents) of their respective attendant variables?

  - Utility? $P(X_1 \mid X_6 = x_6) = \sum_{X_2, X_3} \psi(X_1, X_2, X_3)$

    $P(X_3 \mid X_6 = x_6) = \phi(X_3)$

    $P(x_6) = \sum_{X_4, X_5} \psi(X_4, X_5, x_6)$

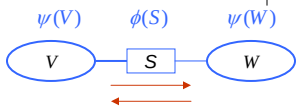


Local operations!

Local Consistency

- We have two ways of obtaining $p(S)$

$$P(S) = \sum_{V \setminus S} \psi(V) \qquad P(S) = \sum_{W \setminus S} \psi(W)$$



and they must be the same

- The following update-rule ensures this:
  - Forward update: $\phi_S^* = \sum_{V \setminus S} \psi_V \qquad \psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W$
  - Backward update $\phi_S^{**} = \sum_{W \setminus S} \psi_W^* \qquad \psi_V^{**} = \frac{\phi_S^{**}}{\phi_S^*} \psi_V^*$
- Two important identities can be proven

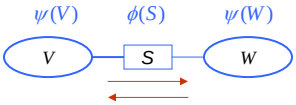
$$\sum_{V \setminus S} \psi_V^{**} = \sum_{W \setminus S} \psi_W^* = \phi_S^{**}$$

Local Consistency

$$\frac{\psi_V^* \psi_W^*}{\phi_S^*} = \frac{\psi_V^{**} \psi_W^{**}}{\phi_S^{**}} = \frac{\psi_V \psi_W}{\phi_S}$$

Invariant Joint

Message Passing Algorithm



$$\phi_S^* = \sum_{V \setminus S} \psi_V \qquad \psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W$$

$$\phi_S^{**} = \sum_{W \setminus S} \psi_W^* \qquad \psi_V^{**} = \frac{\phi_S^{**}}{\phi_S^*} \psi_V^*$$

- This simple local message-passing algorithn on a clique tree defines the general probability propagation algorithm for directed graphs!
  - Many interesting algorithms are special cases:
    - Forward-backward algorithm for hidden Markov models,
    - Kalman filter updates
    - Pealing algorithms for probabilistic trees
  - The algorithm seems reasonable. Is it correct?

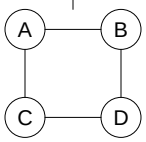
A problem

- Consider the following graph and a corresponding clique tree

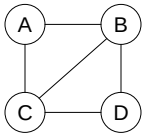
  - Note that C appears in two non-neighboring cliques
- *Question*: with the previous message passage, can we ensure that the probability associated with C in these two (non-neighboring) cliques consistent?
- Answer: No. It is not true that in general local consistency implies global consistency
- What else do we need to get such a guarantee?

Triangulation

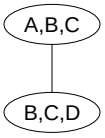
- A triangulated graph is one in which *no cycles* with four or more nodes exist in which there is no *chord*



- We triangulate a graph by adding chords:
- Now we no longer have our global inconsistency problem.



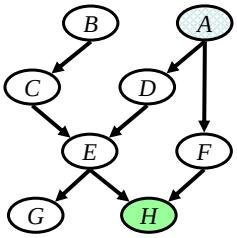
  - A clique tree for a triangulated graph has the *running intersection property*: If a node appears in two cliques, it appears everywhere on the path between the cliques



  - Thus local consistency implies global consistency

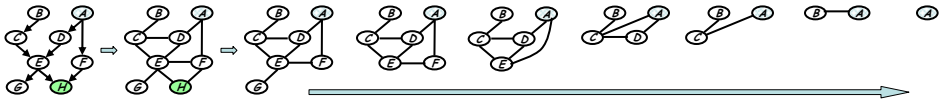
Junction trees

- A clique tree for a triangulated graph is referred to as a *junction tree*
- In junction trees, local consistency implies global consistency. Thus the local message-passing algorithms is (provably) correct
- It is also possible to show that *only* triangulated graphs have the property that their clique trees are junctions. Thus if we want local algorithms, we *must* triangulate
- Are we now all set?
  - How to triangulate?
  - The complexity of building a JT depends on how we triangulate!!
  - Consider this network: it turns out that we will need to pay an $O(2^4)$ or $O(2^6)$ cost depending on how we triangulate!



How to triangulate

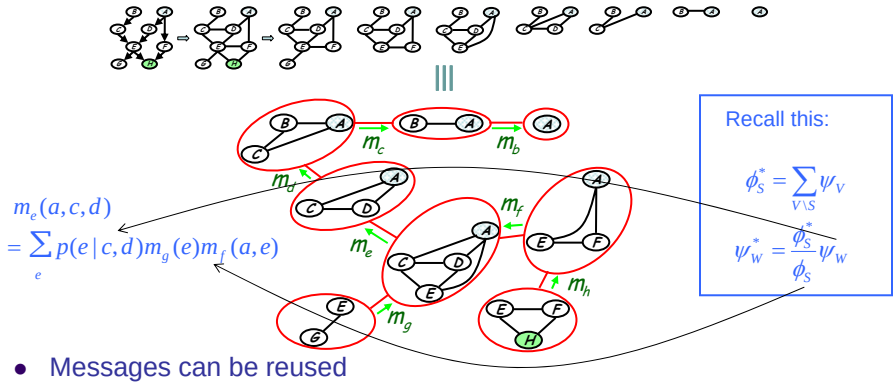
- A graph elimination algorithm



- Intermediate terms correspond to the cliques resulted from elimination
  - "good" elimination orderings lead to **small cliques** and hence reduce complexity (what will happen if we eliminate "e" first in the above graph?)
  - finding the optimum ordering is NP-hard, but for many graph optimum or near-optimum can often be heuristically found

From Elimination to Message Passing

- Our algorithm so far answers only one query (e.g., on one node), do we need to do a complete elimination for every such query?
- Elimination ≡ message passing on a **clique tree**



Recall this:

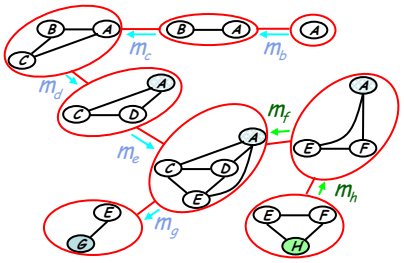
$$\phi_S^* = \sum_{V \setminus S} \psi_V$$

$$\psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W$$

- Messages can be reused

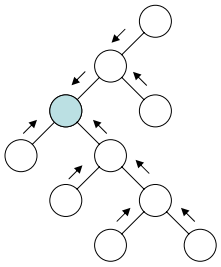
From Elimination to Message Passing

- Our algorithm so far answers only one query (e.g., on one node), do we need to do a complete elimination for every such query?
- Elimination ≡ message passing on a **clique tree**
  - **Another query ...**

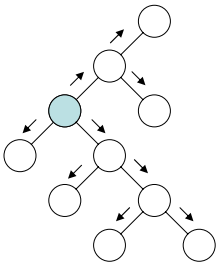


- Messages m_f and m_h are reused, others need to be recomputed

Message-passing algorithms



collect



distribute

- Message update
  - The Hugin update
  - The Shafer-Shenoy update

$$\phi_S^* = \sum_{V \setminus S} \psi_V \qquad \psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W$$

$$m_{i \to j}(S_{ij}) = \sum_{C_i \setminus S_{ij}} \psi_{C_i} \prod_{k \neq j} m_{k \to i}(S_{ki})$$

A Sketch of the Junction Tree Algorithm

- **The algorithm**

1. Moralize the graph (trivial)
2. Triangulate the graph (good heuristic exist, but actually NP hard)
3. Build a clique tree (e.g., using a maximum spanning tree algorithm
4. Propagation of probabilities --- a local message-passing protocol

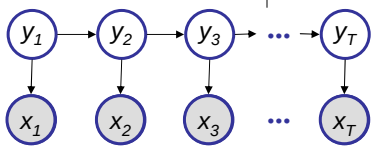
- Results in marginal probabilities of all cliques --- solves all queries in a single run
- A **generic** exact inference algorithm for any GM
- **Complexity**: exponential in the size of the maximal clique --- a good elimination order often leads to small maximal clique, and hence a good (i.e., thin) JT

Case study:

- **Hidden Markov Model**

Recall definition of HMM

- Transition probabilities between any two states



$$p(y_t^j = 1 \mid y_{t-1}^i = 1) = a_{i,j},$$

or $$p(y_t \mid y_{t-1}^i = 1) \sim \text{Multinomial}(a_{i,1}, a_{i,2}, \ldots, a_{i,M}), \forall i \in \mathbb{I}.$$

- Start probabilities

$$p(y_1) \sim \text{Multinomial}(\pi_1, \pi_2, \ldots, \pi_M).$$

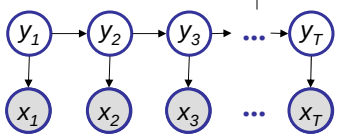
- Emission probabilities associated with each state

$$p(x_t \mid y_t^i = 1) \sim \text{Multinomial}(b_{i,1}, b_{i,2}, \ldots, b_{i,K}), \forall i \in \mathbb{I}.$$

or in general: $$p(x_t \mid y_t^i = 1) \sim \text{f}(\cdot \mid \theta_i), \forall i \in \mathbb{I}.$$

Probability of a parse

- Given a sequence $\mathbf{x} = x_1 \ldots\ldots x_T$ and a parse $\mathbf{y} = y_1, \ldots\ldots, y_T$,
- To find how likely is the parse: (given our HMM and the sequence)



$$
\begin{aligned}
p(\mathbf{x}, \mathbf{y}) &= p(x_1 \ldots\ldots x_T, y_1, \ldots\ldots, y_T) \qquad \text{(Joint probability)} \\
&= p(y_1)\, p(x_1 \mid y_1)\, p(y_2 \mid y_1)\, p(x_2 \mid y_2) \ldots p(y_T \mid y_{T-1})\, p(x_T \mid y_T) \\
&= p(y_1)\, \mathrm{P}(y_2 \mid y_1) \ldots p(y_T \mid y_{T-1}) \times p(x_1 \mid y_1)\, p(x_2 \mid y_2) \ldots p(x_T \mid y_T) \\
&= p(y_1, \ldots\ldots, y_T)\, p(x_1 \ldots\ldots x_T \mid y_1, \ldots\ldots, y_T)
\end{aligned}
$$

Let $\pi_{y_1} \overset{\text{def}}{=} \prod_{i=1}^{M} [\pi_i]^{y_1^i}$, $a_{y_t, y_{t-1}} \overset{\text{def}}{=} \prod_{i,j=1}^{M} [a_{ij}]^{y_t^i y_{t-1}^j}$, and $b_{y_t, x_t} \overset{\text{def}}{=} \prod_{i=1}^{M} \prod_{k=1}^{K} [b_{ik}]^{y_t^i x_t^k}$,

$$= \pi_{y_1} a_{y_1, y_2} \cdots a_{y_{T-1}, y_T}\; b_{y_1, x_1} \cdots b_{y_T, x_T}$$

- Marginal probability: $p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) = \sum_{y_1} \sum_{y_2} \cdots \sum_{y_N} \pi_{y_1} \prod_{t=2}^{T} a_{y_{t-1}, y_t} \prod_{t=1}^{T} p(x_t \mid y_t)$
- Posterior probability: $p(\mathbf{y} \mid \mathbf{x}) = p(\mathbf{x}, \mathbf{y}) / p(\mathbf{x})$

Three main questions on HMMs

1. Evaluation

GIVEN	an HMM ***M***,	and a sequence **x**,
FIND	Prob (**x** | ***M***)	
ALGO.	Forward	

2. Decoding

GIVEN	an HMM ***M***, and a sequence **x** ,
FIND	the sequence **y** of states that maximizes, e.g., P(**y** | **x** , ***M***), or the most probable subsequence of states
ALGO.	Viterbi, Forward-backward

3. Learning (next lecture)

GIVEN	an HMM ***M***, with unspecified transition/emission probs., and a sequence **x**,
FIND	parameters $\theta = (\pi_i, a_{ij}, \eta_{ik})$ that maximize P(**x** | θ)
ALGO.	Baum-Welch (EM)

The Forward Algorithm

- We want to calculate $P(\mathbf{x})$, the likelihood of $\mathbf{x}$, given the HMM
  - Sum over all possible ways of generating $\mathbf{x}$:

$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x},\mathbf{y}) = \sum_{y_1}\sum_{y_2}\cdots\sum_{y_N} \pi_{y_1}\prod_{t=2}^{T} a_{y_{t-1},y_t}\prod_{t=1}^{T} p(x_t \mid y_t)$$

  - To avoid summing over an exponential number of paths $\mathbf{y}$, define

$$\alpha(y_t^k = 1) = \alpha_t^k \overset{\text{def}}{=} P(x_1,\ldots,x_t, y_t^k = 1) \qquad \text{(the forward probability)}$$

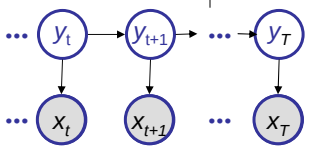
  - The recursion:

$$\alpha_t^k = p(x_t \mid y_t^k = 1)\sum_i \alpha_{t-1}^i a_{i,k}$$

$$P(\mathbf{x}) = \sum_k \alpha_T^k$$

The Backward Algorithm

- We want to compute $P(y_t^k = 1 \mid \mathbf{x})$, the posterior probability distribution on the t^{th} position, given $\mathbf{x}$



  - We start by computing

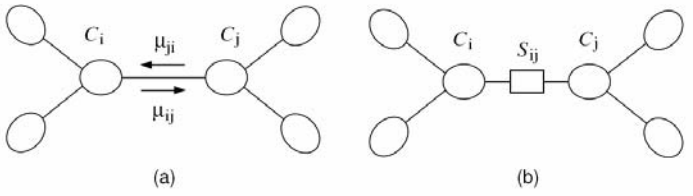
$$P(y_t^k = 1, \mathbf{x}) = P(x_1,\ldots,x_t, y_t^k = 1, x_{t+1},\ldots,x_T)$$
$$= P(x_1,\ldots,x_t, y_t^k = 1)P(x_{t+1},\ldots,x_T \mid x_1,\ldots,x_t, y_t^k = 1)$$
$$= P(x_1\ldots x_t, y_t^k = 1)P(x_{t+1}\ldots x_T \mid y_t^k = 1)$$

Forward, α_t^k **Backward,** $\beta_t^k = P(x_{t+1},\ldots,x_T \mid y_t^k = 1)$

  - The recursion: $\beta_t^k = \sum_i a_{k,i}\, p(x_{t+1} \mid y_{t+1}^i = 1)\beta_{t+1}^i$

Shafer Shenoy for HMMs

- Recap: Shafer-Shenoy algorithm



  - Message from clique i to clique j:

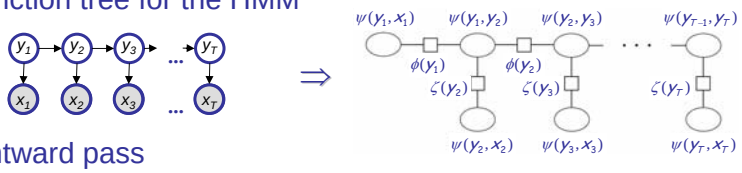
$$\mu_{i\to j} = \sum_{C_i \setminus S_{ij}} \psi_{C_i} \prod_{k\neq j} \mu_{k\to i}(S_{ki})$$

  - Clique marginal

$$p(C_i) \propto \psi_{C_i} \prod_{k} \mu_{k\to i}(S_{ki})$$

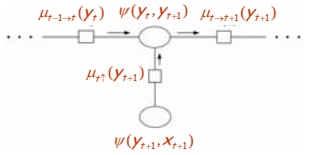
Shafer Shenoy for HMMs (cont.)

- A junction tree for the HMM



- Rightward pass

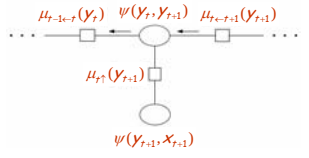
$$\mu_{t\to t+1}(y_{t+1}) = \sum_{y_t} \psi(y_t, y_{t+1}) \mu_{t-1\to t}(y_t) \mu_{t\uparrow}(y_{t+1})$$
$$= \sum_{y_t} p(y_{t+1} \mid y_t) \mu_{t-1\to t}(y_t) p(x_{t+1} \mid y_{t+1})$$
$$= p(x_{t+1} \mid y_{t+1}) \sum_{y_t} a_{y_t, y_{t+1}} \mu_{t-1\to t}(y_t)$$



  - This is exactly the *forward algorithm*!

- Leftward pass …

$$\mu_{t-1\leftarrow t}(y_t) = \sum_{y_{t+1}} \psi(y_t, y_{t+1}) \mu_{t\leftarrow t+1}(y_{t+1}) \mu_{t\uparrow}(y_{t+1})$$
$$= \sum_{y_{t+1}} p(y_{t+1} \mid y_t) \mu_{t\leftarrow t+1}(y_{t+1}) p(x_{t+1} \mid y_{t+1})$$



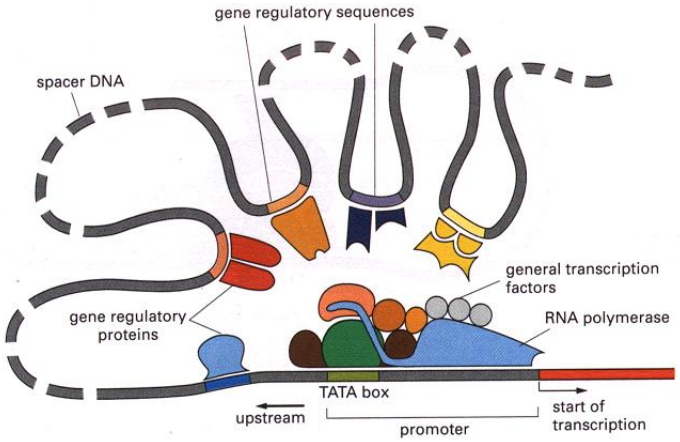
  - This is exactly the *backward algorithm*!

Approaches to inference

- Exact inference algorithms
  - The elimination algorithm
  - The junction tree algorithms √
  -
- Approximate inference techniques
  - Stochastic simulation / sampling methods
  - Markov chain Monte Carlo methods
  - Variational algorithms (later lectures) √

The motif detection problem

Biological background: the transcriptional regulatory machinery



In silico motif detection

multiple alignment: $\mathbf{A}=$

```
1: AAAAGAGTCA
2: AAATGACTCA
.  AAGTGAGTCA
.  AAAAGAGTCA
.  GGATGAGTCA
.  AAATGAGTCA
.  GAATGAGTCA
M: AAAAGAGTCA
```


A Generative Scheme

$\mathbf{A}=$

```
1: AAAAGAGTCA
2: AAATGACTCA
.  AAGTGAGTCA
.  AAAAGAGTCA
.  GGATGAGTCA
.  AAATGAGTCA
.  GAATGAGTCA
M: AAAAGAGTCA
```

Locations: $\{X\}$ Background

$\{Y\}=$

```
5′- ATGGCAGAATCACTTTAAAACGTGGCCCCACCCGCTGCACCCTGTGCATTTTGTACGTTACTGCGAAATGACTCAACG
5′- CACATCCAACGAATCACCTCACCGTTATCGTGACTCACTTTCTTTCGCATCGCCGAAGTGCCATAAAAAATATTTTTT
5′- TGCGAACAAAAGAGTCATTACAACGAGGAAATAGAAGAAAATGAAAAATTTTCGACAAAATGTATAGTCATTTCTATC
5′- ACAAAGGTACCTTCCTGGCCAATCTCACAGATTTAATATAGTAAATTGTCATGCATATGACTCATCCCGAACATGAAA
5′- ATTGATTGACTCATTTTCCTCTGACTACTACCAGTTCAAAATGTTAGAGAAAAATAGAAAAGCAGAAAAAATAAATAA
5′- GGCGCCACAGTCCGCGTTTGGTTATCCGGCTGACTCATTCTGACTCTTTTTTGGAAAGTGTGGCATGTGCTTCACACA
```

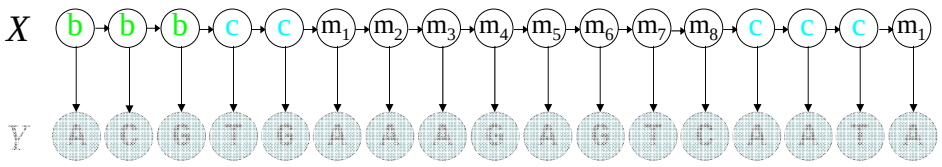
The LOGOS motif model

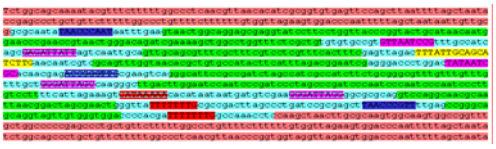
[Xing, Wu, Jordan and Karp, JBCB 2004]

- An integrated LOcal and GlObal motif Sequence model
  - The occurrences of motifs in DNA sequences are governed by a global motif syntax model $p_g(X)$
  - All instances of a specific motif admit a local motif prior model $p_l(\mathbf{A}, \theta | X)$
  - Non-motif sequences are modeled by a background model $p_b(Y\text{-}\mathbf{A}|X)$

The global hidden Markov model (HMM)

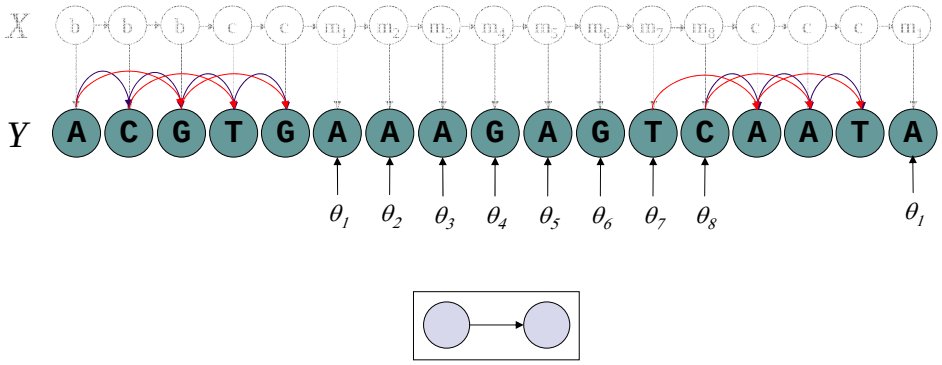
1st order Markov grammar





The background model

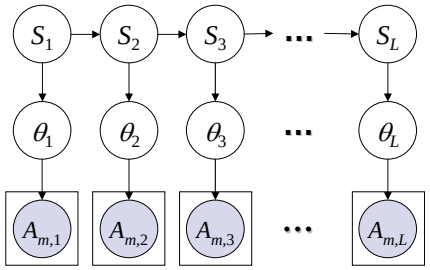
k-th order Markov background



The local prior model

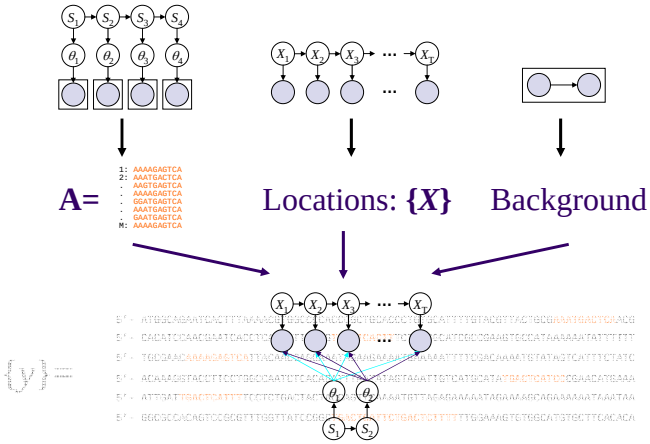
Hidden Markov Dicichlet-multinomial (HMDM)

[Xing, Jordan, Karp and Russell, NIPS 2002]



AAAAGAGTCA
AAAAGAGTCA
AAATGACTCA
AAGTGAGTCA
AAAAGAGTCA
GGATGAGTCA
AAATGAGTCA
GAATGAGTCA
AAAAGAGTCA

A modular Bayesian model for motif detection



Joint posterior:

$$p(X, S, \theta \mid \mathbf{y}) = p(X \mid \theta, \mathbf{y})\, p(\theta, S)$$

Inference in LOGOS model

- Joint posterior: $p(X, S, \theta \mid \mathbf{y}) = p(X \mid \theta, \mathbf{y})\, p(\theta, S)$
  - inference on motif locations

$$p(x_t \mid \mathbf{y}) = \int_{\theta} \sum_{s} \sum_{\forall x_{t' \neq t}} p(\mathbf{x} \mid \theta, \mathbf{y})\, p(\theta, s)$$

  - state space to be summed (and integrated) over

$$\Re^{4 \times \sum L_k} \times |\Omega_s|^{\sum L_k} \times |\Omega_x|^{T}$$

$\sim \Re^{120} \times 10^{1200}$ for a 1000bp sequence with two motif patterns of length 15bp

- Approximate inference
  - Stochastic approximation: Gibbs sampling
  - Deterministic approximation: Variational inference √

Variational Methods

- 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:

$$\text{e.g.} \quad f^* = \arg\max_{f \in \mathcal{S}} \{ F(f) \}$$

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

Exponential Family

- Exponential representation of graphical models:

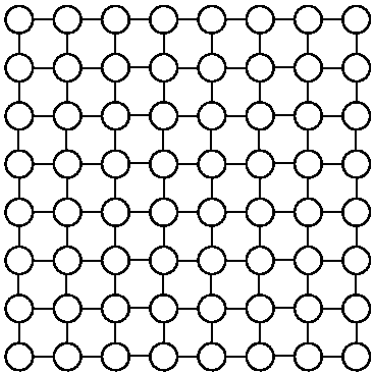
$$P(\mathbf{X}) = \frac{1}{Z}\prod_{c\in C}\psi_c(\mathbf{X}_c) \quad \Rightarrow \quad p(\mathbf{X}|\boldsymbol{\theta}) = \exp\left\{\sum_{\alpha}\theta_\alpha\phi_\alpha(\mathbf{X}_{D_\alpha}) - A(\boldsymbol{\theta})\right\}$$

- Includes discrete models, Gaussian, Poisson, exponential, and many others

$E(\mathbf{X}) = -\sum_{\alpha}\theta_\alpha\phi_\alpha(\mathbf{X}_{D_\alpha})$ is referred to as the *energy* of state $\mathbf{x}$

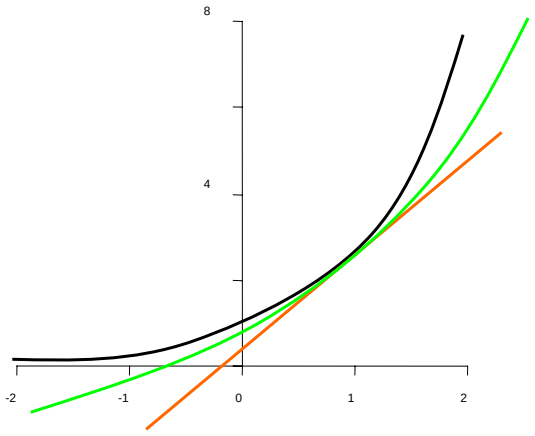
$$\Rightarrow \quad p(\mathbf{X}|\boldsymbol{\theta}) = \exp\{-E(\mathbf{X}) - A(\boldsymbol{\theta})\}$$
$$= \exp\{-E(\mathbf{X}_H, \mathbf{x}_E) - A(\boldsymbol{\theta}, \mathbf{x}_E)\}$$

Example: the Boltzmann distribution on atomic lattice



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

Lower bounds of exponential functions



$$\exp(x) \geq \exp(\mu)(x-\mu+1)$$

$$\exp(x) \geq \frac{1}{6}\exp(\mu)\left((x-\mu)^3 + 3(x-\mu)^2 + 6(x-\mu+1)\right)$$

Lower bounding likelihood

Representing $q(\mathbf{X}_H)$ by $\exp\{-E'(\mathbf{X}_H)\}$:

Lemma: Every marginal distribution $q(\mathrm{X}_H)$ defines a lower bound of likelihood:

$$p(\mathbf{x}_E) \geq \int d\mathbf{x}_H \exp\{-E'(\mathbf{x}_H)\} \left(1 - A(\mathbf{x}_E) - \left(E(\mathbf{x}_H, \mathbf{x}_E) - E'(\mathbf{x}_H)\right)\right),$$

where x_E denotes observed variables (evidence).

Upgradeable to higher order bound [Leisink and Kappen, 2000]

Lower bounding likelihood

Representing $q(\mathbf{X}_H)$ by $\exp\{-E'(\mathbf{X}_H)\}$:

Lemma: Every marginal distribution $q(\mathrm{X}_H)$ defines a lower bound of likelihood:

$$p(\mathbf{x}_E) \geq C - \left\langle E(\mathbf{X}_H, \mathbf{x}_E) \right\rangle_{q(\mathbf{X}_H)} + \int d\mathbf{x}_H q(\mathbf{x}_H) \log q(\mathbf{x}_H)$$
$$= C - \left\langle E \right\rangle_q - H_q,$$

where x_E denotes observed variables (evidence).

$\left\langle E \right\rangle_q$: expected energy $\quad$ $\left\langle E \right\rangle_q + H_q$: Gibbs free energy

H_q : entropy

KL and variational (Gibbs) free energy

- Kullback-Leibler Distance:

$$KL(q \,\|\, p) \equiv \sum_z q(z) \ln \frac{q(z)}{p(z)}$$

- "Boltzmann's Law" (definition of "energy"):

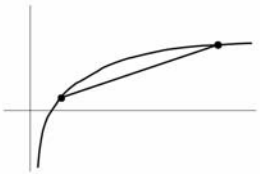
$$p(z) = \frac{1}{C} \exp\left[-E(z)\right]$$

$$KL(q \,\|\, p) \equiv \underbrace{\sum_z q(z)E(z) + \sum_z q(z) \ln q(z)}_{\text{Gibbs Free Energy } G(q);\ \text{minimized when } q(Z) = p(Z)} + \ln C$$

KL and Log Likelihood

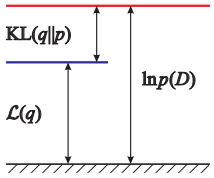
- Jensen's inequality

$$\begin{aligned}
\ell(\theta; x) &= \log p(x \mid \theta) \\
&= \log \sum_z p(x, z \mid \theta) \\
&= \log \sum_z q(z \mid x) \frac{p(x, z \mid \theta)}{q(z \mid x)} \\
&\geq \sum_z q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)}
\end{aligned} \quad \Rightarrow \quad \ell(\theta; x) \geq \left\langle \ell_c(\theta; x, z) \right\rangle_q + H_q = \mathcal{L}(q)$$



- KL and Lower bound of likelihood

$$\begin{aligned}
\ell(\theta; x) = \log p(x \mid \theta) &= \log \frac{p(x, z \mid \theta)}{p(z \mid x, \theta)} = \sum_z q(z) \log \frac{p(x, z \mid \theta)}{p(z \mid x, \theta)} \\
&= \sum_z q(z) \log \frac{p(x, z \mid \theta)}{q(z)} \frac{q(z)}{p(z \mid x, \theta)} \\
&= \sum_z q(z) \log \frac{p(x, z \mid \theta)}{q(z)} + \sum_z q(z) \log \frac{q(z)}{p(z \mid x, \theta)}
\end{aligned} \quad \Rightarrow \quad \ell(\theta; x) = \mathcal{L}(q) + KL(q \,\|\, p)$$



- Setting $q()$=$p(z|x)$ closes the gap (c.f. EM)

A variational representation of probability distributions

$$q = \arg\max_{q \in Q} \left\{ -\langle E \rangle_q - H_q \right\}$$
$$= \arg\min_{q \in Q} \left\{ \langle E \rangle_q + H_q \right\}$$

where Q is the equivalent sets of realizable distributions, *e.g.*, all valid parameterizations of exponential family distributions, marginal polytopes [winright *et al.* 2003].

Difficulty: H_q is intractable for general q

"solution": approximate H_q
and/or,
relax or tighten Q

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: $Q \rightarrow \mathcal{T} \quad (\mathcal{T} \subseteq Q)$

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

Belief Propagation

- Do not optimize $q(\mathbf{X}_H)$ explicitly, but 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: $\mathcal{M}_o = \{\ \tau \geq 0 \mid \sum_{x_i} \tau(x_i) = 1, \sum_{x_i} \tau(x_i, x_j) = \tau(x_j)\ \}$

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

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

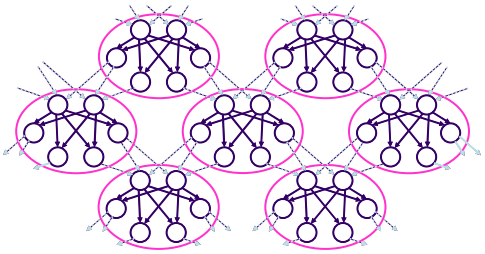
Mean Field Approximation

Cluster-based approx. to the Gibbs free energy

(Wiegerinck 2001, Xing *et al* 03,04)

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

Clusters: $G[\{q_c(X_c)\}]$



Mean field approx. to Gibbs free energy

- Given a disjoint clustering, $\{C_1, \ldots, C_l\}$, of all variables
- Let $$q(\mathbf{X}) = \prod_i q_i(\mathbf{X}_{C_i}),$$
- Mean-field free energy

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

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

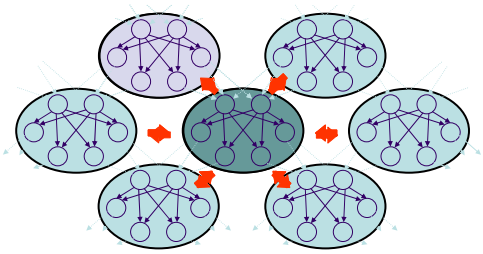
The Generalized Mean Field theorem

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:

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

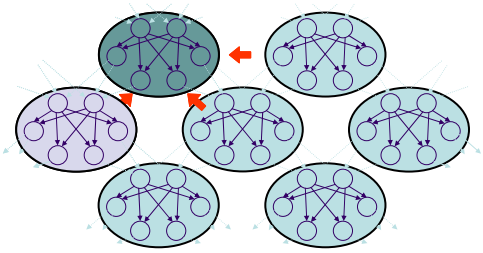
GMF algorithm: Iterate over each q_i

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



A generalized mean field algorithm

[xing *et al*. UAI 2003]



Convergence theorem

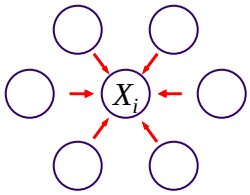
Theorem: The GMF algorithm is guaranteed to converge to a local optimum, and provides a lower bound for the likelihood of evidence (or partition function) the model.

The naive mean field approximation

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

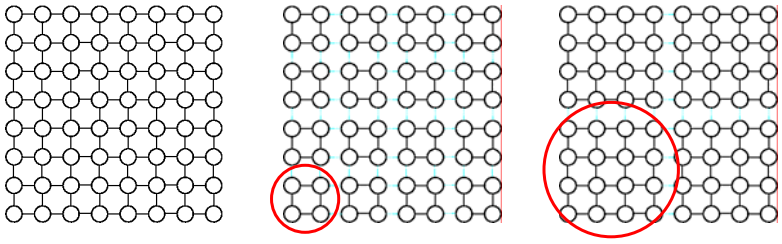
mean field equation:

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



- $\left\langle X_j\right\rangle_{q_j}$ 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

Generalized MF approximation to Ising models

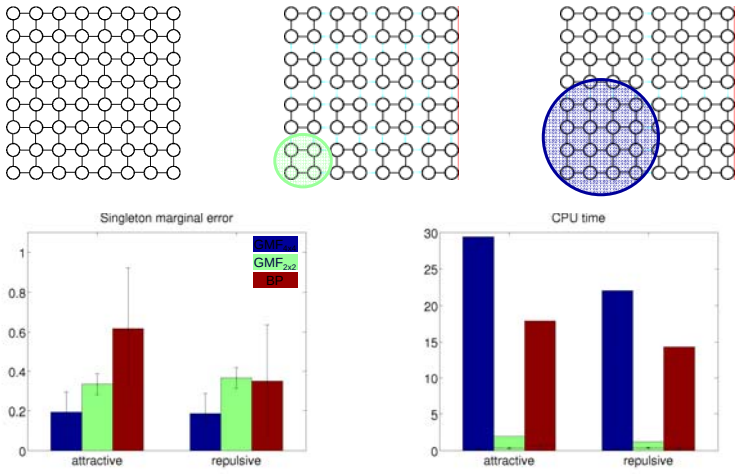


Cluster marginal of a square block C_k:

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

Virtually a reparameterized Ising model of small size.

GMF approximation to Ising models



Attractive coupling: positively weighted
Repulsive coupling: negatively weighted

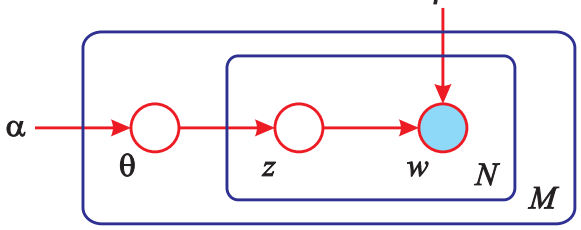
Cluster-based MF (e.g., GMF)

- 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 1: Latent Dirichlet Allocation

- Blei, Jordan and Ng (2003)
- Generative model of documents (but broadly applicable e.g. collaborative filtering, image retrieval, bioinformatics)
- Generative model:
  - choose
  - choose topic
  - choose word

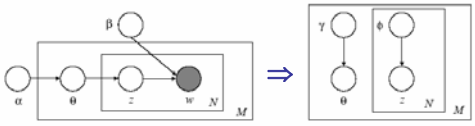
$$\theta \sim \mathrm{Dir}(\alpha)$$
$$\mathbf{z}_n \sim \mathrm{Mult}(\theta)$$
$$\mathbf{w}_n \sim p(\mathbf{w}_n|\mathbf{z}_n, \beta)$$



Latent Dirichlet Allocation

- Variational approximation

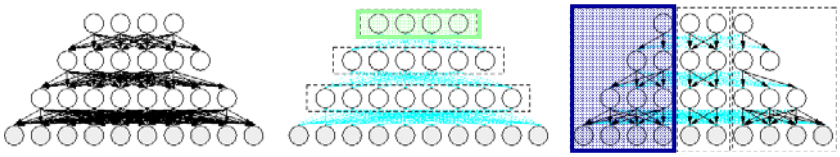
$$q(\theta, \mathbf{z}) = q_\theta(\theta) q_{\mathbf{z}}(\mathbf{z})$$
$$= \mathrm{Dir}(\theta \mid \gamma = f(\alpha, \langle \mathbf{z} \rangle)) \times$$
$$\mathrm{Multi}(\mathbf{z} \mid \phi = f(\beta_w, \langle \ln \theta \rangle))$$

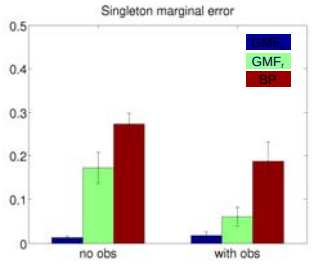


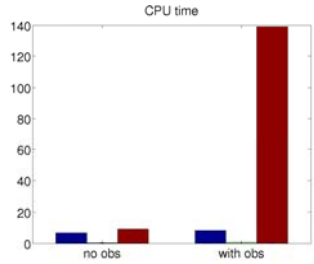
$$\phi_{ni} \propto \beta_{iw_n} \exp\{\mathrm{E}_q[\log(\theta_i) \mid \gamma]\}$$
$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni}.$$

- Data set:
  - 15,000 documents
  - 90,000 terms
  - 2.1 million words
- Model:
  - 100 factors
  - 9 million parameters
- MCMC could be totally infeasible for this problem

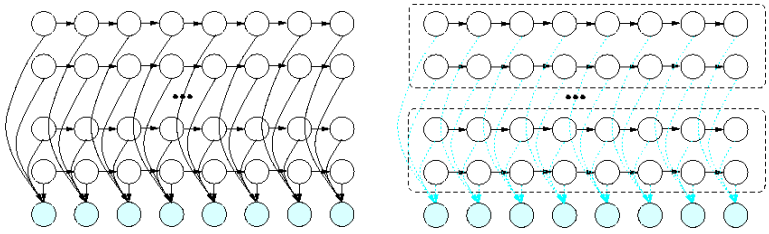
Example 2: Sigmoid belief network

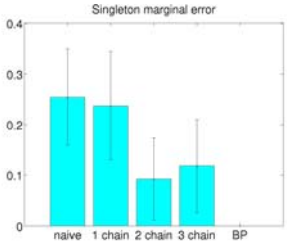


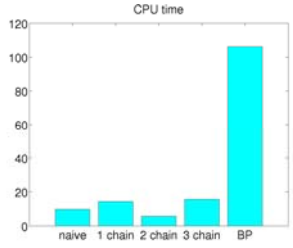




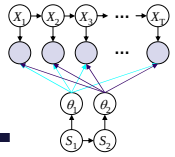
Example 3: Factorial HMM





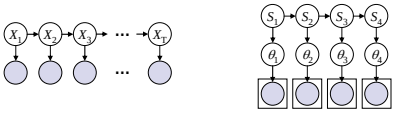


Example 4: GMF approximation to LOGOS



- Approximate $p(X,S,\theta|y)$ with a tractable distribution $q(X,S,\theta)$
- Variable partition:

$$\{X,S,\theta\} \quad = \quad \{X\} \quad + \quad \{S,\theta\}$$



- Let

$$q(X,S,\theta) = q_1(X)q_2(\theta,S)$$

GMF for DNA motif prediction

- GMF approximations: $q(X,S,\theta) \propto q_1(X)q_2(\theta,S)$

$$q_1^*(X) = p(X \mid y, \tilde{\theta})$$
$$q_2^*(\theta,S) = p(S)p(\theta \mid S, \tilde{\mathbf{A}}_{Y|X})$$

- GMF algorithm

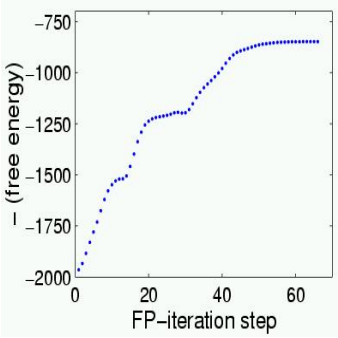
Compute expectation of **A** w.r.t. *local* posterior of x in ***HMM submodel*** assuming θ is given

$\tilde{\mathbf{A}}_{Y|X}$

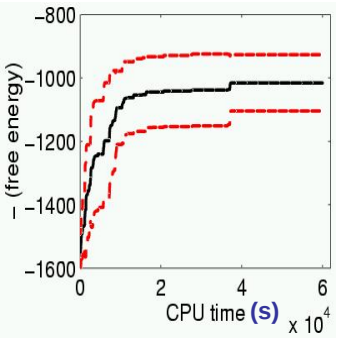
$\tilde{\theta}$

Compute Bayesian est. of θ, w.r.t its *local* posterior in ***HMDM submodel*** assuming **A** is given

Traces of GMF iterations

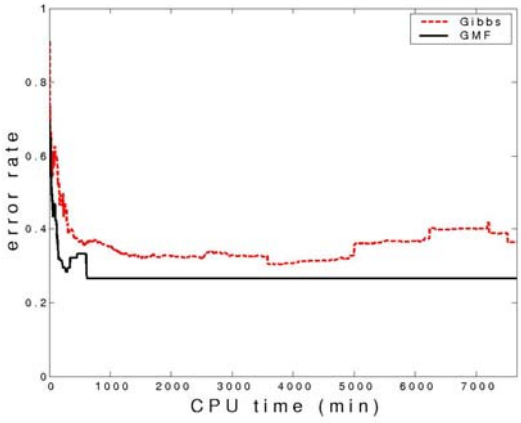


A single round of FP-iteration



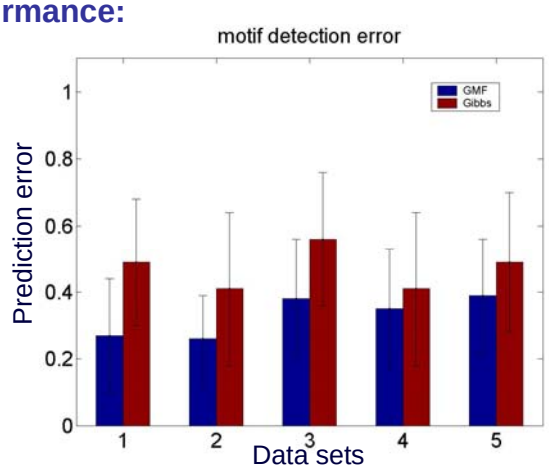
Sequentialized multiple random restarts

GMF vs. Gibbs sampler on motif detection



GMF vs. Gibbs sampler on motif detection

Performance:



Sampling time of Gibbs = 10× the time for GMF

Open Problem

- Idea:
  - A(**θ**) is convex
    - Epigraph of A(**θ**) can be represented as a pointwise supremum of all affine functions that are global under-estimators of A(**θ**)
    - Variationally, compute A(**θ**) using the following convex optimization:

$$A(\theta) = \sup_{\mu \in \mathbb{R}^d} \{\langle \theta, \mu \rangle - A^*(\mu)\}$$

    - Investigate the form of the dual function $A^*(\boldsymbol{\mu})$

- Important consequence
  - Solution also yields the marginal probabilities!

Martin Wainwright and Michael Jordan
IEEE Transactions on Signal Processing, 2006