All of Graphical Models

Xiaojin Zhu

Department of Computer Sciences
University of Wisconsin–Madison, USA

Tutorial at ICMLA 2011
Given GM = joint distribution $p(x_1, \ldots, x_n)$

Do inference = $p(X_Q \mid X_E)$, in general

$X_Q \cup X_E \subset \{x_1 \ldots x_n\}$

If $p(x_1, \ldots, x_n)$ not given, estimate it from data
Outline

Life without Graphical Models

Representation
- Directed Graphical Models (Bayesian Networks)
- Undirected Graphical Models (Markov Random Fields)

Inference
- Exact Inference
- Markov Chain Monte Carlo
- Variational Inference
  - Loopy Belief Propagation
  - Mean Field Algorithm
  - Exponential Family
- Maximizing Problems

Parameter Learning

Structure Learning
Outline

Life without Graphical Models

Representation
  Directed Graphical Models (Bayesian Networks)
  Undirected Graphical Models (Markov Random Fields)

Inference
  Exact Inference
  Markov Chain Monte Carlo
  Variational Inference
    Loopy Belief Propagation
    Mean Field Algorithm
    Exponential Family
  Maximizing Problems

Parameter Learning

Structure Learning
Life without Graphical Models

... is fine mathematically:

- The universe is reduced to a set of random variables $x_1, \ldots, x_n$
  - e.g., $x_1, \ldots, x_{n-1}$ can be the discrete or continuous features
  - e.g., $x_n \equiv y$ can be the discrete class label
- The joint $p(x_1, \ldots, x_n)$ completely describes how the universe works
- “Machine learning”: estimate $p(x_1, \ldots, x_n)$ from training data $X^{(1)}, \ldots, X^{(N)}$, where $X^{(i)} = (x_1^{(i)}, \ldots, x_n^{(i)})$
- “Prediction”: $y^* = \text{argmax } p(x_n \mid x_1^*, \ldots, x_n^*)$, a.k.a. inference
  - by the definition of conditional probability

$$p(x_n \mid x_1^*, \ldots, x_n^*) = \frac{p(x_1^*, \ldots, x_n^*, x_n)}{\sum_v p(x_1^*, \ldots, x_n^*, x_n = v)}$$
Conclusion

- Life without graphical models is just fine
- So why are we still here?
Given GM = joint distribution $p(x_1, \ldots, x_n)$
- exponential naïve storage ($2^n$ for binary r.v.)
- hard to interpret (conditional independence)

Do inference = $p(X_Q \mid X_E)$, in general
$X_Q \cup X_E \subseteq \{x_1 \ldots x_n\}$
- Often can’t do it computationally

If $p(x_1, \ldots, x_n)$ not given, estimate it from data
- Can’t do it either
Acknowledgments Before We Start

Much of this tutorial is based on

Outline

Life without Graphical Models

Representation
  Directed Graphical Models (Bayesian Networks)
  Undirected Graphical Models (Markov Random Fields)

Inference
  Exact Inference
  Markov Chain Monte Carlo
  Variational Inference
    Loopy Belief Propagation
    Mean Field Algorithm
    Exponential Family
  Maximizing Problems

Parameter Learning

Structure Learning
“Graphical model” is the study of probabilistic models.

Just because there is a graph with nodes and edges doesn’t mean it’s GM.

These are not graphical models.

- Neural network
- Decision tree
- Network flow
- HMM template
Outline

Life without Graphical Models

Representation
- Directed Graphical Models (Bayesian Networks)
- Undirected Graphical Models (Markov Random Fields)

Inference
- Exact Inference
- Markov Chain Monte Carlo
- Variational Inference
  - Loopy Belief Propagation
  - Mean Field Algorithm
  - Exponential Family
- Maximizing Problems

Parameter Learning

Structure Learning
Bayesian Network

- A directed graph has nodes $X = (x_1, \ldots, x_n)$, some of them connected by directed edges $x_i \rightarrow x_j$
- A cycle is a directed path $x_1 \rightarrow \ldots \rightarrow x_k$ where $x_1 = x_k$
- A directed acyclic graph (DAG) contains no cycles
- A Bayesian network on the DAG is a family of distributions satisfying
  \[ \{ p \mid p(X) = \prod_{i} p(x_i \mid Pa(x_i)) \} \]
  where $Pa(x_i)$ is the set of parents of $x_i$.
- $p(x_i \mid Pa(x_i))$ is the conditional probability distribution (CPD) at $x_i$
- By specifying the CPDs for all $i$, we specify a particular distribution $p(X)$
Example: Alarm

Binary variables

\[ P(B, \sim E, A, J, \sim M) \]
\[ = P(B)P(\sim E)P(A \mid B, \sim E)P(J \mid A)P(\sim M \mid A) \]
\[ = 0.001 \times (1 - 0.002) \times 0.94 \times 0.9 \times (1 - 0.7) \]
\[ \approx .000253 \]
Example: Naive Bayes

\[ p(y, x_1, \ldots, x_d) = p(y) \prod_{i=1}^{d} p(x_i \mid y) \]

- Used extensively in natural language processing
- Plate representation on the right
No Causality Whatsoever

The two BNs are equivalent in all respects:

- Bayesian networks imply no causality at all
- They only encode the joint probability distribution (hence correlation)
- However, people tend to design BNs based on causal relations
Example: Latent Dirichlet Allocation (LDA)

A generative model for $p(\phi, \theta, z, w \mid \alpha, \beta)$:

For each topic $t$

$\phi_t \sim \text{Dirichlet}(\beta)$

For each document $d$

$\theta \sim \text{Dirichlet}(\alpha)$

For each word position in $d$

- topic $z \sim \text{Multinomial}(\theta)$
- word $w \sim \text{Multinomial}(\phi_z)$

Inference goals: $p(z \mid w, \alpha, \beta)$, $\arg\max_{\phi,\theta} p(\phi, \theta \mid w, \alpha, \beta)$
A generative model for $p(\phi, \theta, z, w \mid \alpha, \beta)$:
For each topic $t$

$$\phi_t \sim \text{Dirichlet}(\beta)$$

For each document $d$

$$\theta \sim \text{Dirichlet}(\alpha)$$

For each word position in $d$

- topic $z \sim \text{Multinomial}(\theta)$
- word $w \sim \text{Multinomial}(\phi_z)$

Inference goals: $p(z \mid w, \alpha, \beta), \arg\max_{\phi, \theta} p(\phi, \theta \mid w, \alpha, \beta)$
A generative model for \( p(\phi, \theta, z, w | \alpha, \beta) \):

For each topic \( t \)
\[
\phi_t \sim \text{Dirichlet}(\beta)
\]

For each document \( d \)
\[
\theta \sim \text{Dirichlet}(\alpha)
\]

For each word position in \( d \)
\[
\text{topic } z \sim \text{Multinomial}(\theta) \\
\text{word } w \sim \text{Multinomial}(\phi_z)
\]

Inference goals:
\[
p(z | w, \alpha, \beta), \arg\max_{\phi, \theta} p(\phi, \theta | w, \alpha, \beta)
\]
A generative model for $p(\phi, \theta, z, w \mid \alpha, \beta)$:
For each topic $t$
\[ \phi_t \sim \text{Dirichlet}(\beta) \]
For each document $d$
\[ \theta \sim \text{Dirichlet}(\alpha) \]
For each word position in $d$
\[ \text{topic } z \sim \text{Multinomial}(\theta) \]
\[ \text{word } w \sim \text{Multinomial}(\phi_z) \]
Inference goals: $p(z \mid w, \alpha, \beta), \arg\max_{\phi, \theta} p(\phi, \theta \mid w, \alpha, \beta)$
A generative model for $p(\phi, \theta, z, w \mid \alpha, \beta)$:

For each topic $t$

$\phi_t \sim \text{Dirichlet}(\beta)$

For each document $d$

$\theta \sim \text{Dirichlet}(\alpha)$

For each word position in $d$

topic $z \sim \text{Multinomial}(\theta)$

word $w \sim \text{Multinomial}(\phi_z)$

Inference goals: $p(z \mid w, \alpha, \beta), \arg\max_{\phi,\theta} p(\phi, \theta \mid w, \alpha, \beta)$
A generative model for $p(\phi, \theta, z, w \mid \alpha, \beta)$:

For each topic $t$

$$\phi_t \sim \text{Dirichlet}(\beta)$$

For each document $d$

$$\theta \sim \text{Dirichlet}(\alpha)$$

For each word position in $d$

- topic $z \sim \text{Multinomial}(\theta)$
- word $w \sim \text{Multinomial}(\phi_z)$

Inference goals: $p(z \mid w, \alpha, \beta)$, $\arg\max_{\phi, \theta} p(\phi, \theta \mid w, \alpha, \beta)$
Example: Latent Dirichlet Allocation (LDA)

A generative model for $p(\phi, \theta, z, w \mid \alpha, \beta)$:

For each topic $t$

$\phi_t \sim \text{Dirichlet}(\beta)$

For each document $d$

$\theta \sim \text{Dirichlet}(\alpha)$

For each word position in $d$

- topic $z \sim \text{Multinomial}(\theta)$
- word $w \sim \text{Multinomial}(\phi_z)$

Inference goals: $p(z \mid w, \alpha, \beta), \arg\max_{\phi, \theta} p(\phi, \theta \mid w, \alpha, \beta)$
A generative model for $p(\phi, \theta, z, w \mid \alpha, \beta)$:

For each topic $t$
\[
\phi_t \sim \text{Dirichlet}(\beta)
\]

For each document $d$
\[
\theta \sim \text{Dirichlet}(\alpha)
\]

For each word position in $d$
\[
topic \ z \sim \text{Multinomial}(\theta)\\
word \ w \sim \text{Multinomial}(\phi_z)
\]

Inference goals: $p(z \mid w, \alpha, \beta), \arg\max_{\phi, \theta} p(\phi, \theta \mid w, \alpha, \beta)$
Some Topics by LDA on the Wish Corpus

\[ p(\text{word} \mid \text{topic}) \]

“troops”

“election”

“love”
Conditional Independence

- Two r.v.s A, B are independent if $P(A, B) = P(A)P(B)$ or $P(A|B) = P(A)$ (the two are equivalent)
- Two r.v.s A, B are conditionally independent given C if
  $P(A, B | C) = P(A | C)P(B | C)$ or
  $P(A | B, C) = P(A | C)$ (the two are equivalent)
- This extends to groups of r.v.s
- Conditional independence in a BN is precisely specified by d-separation ("directed separation")
d-Separation Case 1: Tail-to-Tail

- A, B in general dependent
- A, B conditionally independent given C
- C is a tail-to-tail node, blocks the undirected path A-B
d-Separation Case 2: Head-to-Tail

- A, B in general dependent
- A, B conditionally independent given C
- C is a head-to-tail node, blocks the path A-B
A, B in general independent

A, B conditionally dependent given C, or any of C’s descendants

C is a head-to-head node, unblocks the path A-B
Any groups of nodes A and B are conditionally independent given another group C, if all undirected paths from any node in A to any node in B are \textit{blocked}.

A path is blocked if it includes a node $x$ such that either

- The path is head-to-tail or tail-to-tail at $x$ and $x \in C$, or
- The path is head-to-head at $x$, and neither $x$ nor any of its descendants is in $C$. 
d-Separation Example 1

- The path from A to B not blocked by either E or F
- A, B dependent given C
The path from A to B is blocked both at E and F
A, B conditionally independent given F
Outline

Life without Graphical Models

Representation
- Directed Graphical Models (Bayesian Networks)
- Undirected Graphical Models (Markov Random Fields)

Inference
- Exact Inference
- Markov Chain Monte Carlo
- Variational Inference
  - Loopy Belief Propagation
  - Mean Field Algorithm
  - Exponential Family
- Maximizing Problems

Parameter Learning

Structure Learning
The efficiency of directed graphical model (acyclic graph, locally normalized CPDs) also makes it restrictive.

A clique $C$ in an undirected graph is a fully connected set of nodes (note: full of loops!)

Define a nonnegative potential function $\psi_C : X_C \rightarrow \mathbb{R}_+$

An undirected graphical model (aka Markov Random Field) on the graph is a family of distributions satisfying

$$\left\{ p \mid p(X) = \frac{1}{Z} \prod_C \psi_C(X_C) \right\}$$

$Z = \int \prod_C \psi_C(X_C) dX$ is the partition function
Example: A Tiny Markov Random Field

- $x_1, x_2 \in \{-1, 1\}$
- A single clique $\psi_C(x_1, x_2) = e^{ax_1 x_2}$
- $p(x_1, x_2) = \frac{1}{Z} e^{ax_1 x_2}$
- $Z = (e^a + e^{-a} + e^{-a} + e^a)$
- $p(1, 1) = p(-1, -1) = e^a/(2e^a + 2e^{-a})$
- $p(-1, 1) = p(1, -1) = e^{-a}/(2e^a + 2e^{-a})$
- When the parameter $a > 0$, favor homogeneous chains
- When the parameter $a < 0$, favor inhomogeneous chains
Log Linear Models

- Real-valued feature functions \( f_1(X), \ldots, f_k(X) \)
- Real-valued weights \( w_1, \ldots, w_k \)

\[
p(X) = \frac{1}{Z} \exp \left( - \sum_{i=1}^{k} w_i f_i(X) \right)
\]
This is an undirected model with $x \in \{0, 1\}$.

$$p_\theta(x) = \frac{1}{Z} \exp \left( \sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t \right)$$

- $f_s(X) = x_s$, $f_{st}(X) = x_s x_t$
- $w_s = -\theta_s$, $w_{st} = -\theta_{st}$
Example: Image Denoising

\[ \text{argmax}_X P(X|Y) \]

[From Bishop PRML]
Example: Gaussian Random Field

\[ p(X) \sim N(\mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp \left( -\frac{1}{2}(X - \mu)^\top \Sigma^{-1}(X - \mu) \right) \]

- Multivariate Gaussian
- The \( n \times n \) covariance matrix \( \Sigma \) positive semi-definite
- Let \( \Omega = \Sigma^{-1} \) be the precision matrix
- \( x_i, x_j \) are conditionally independent given all other variables, if and only if \( \Omega_{ij} = 0 \)
- When \( \Omega_{ij} \neq 0 \), there is an edge between \( x_i, x_j \)
Two group of variables A, B are conditionally independent given another group C, if

- Remove C and all edges involving C
- A, B become disconnected
Factor Graph

- For both directed and undirected graphical models
- Bipartite: edges between a variable node and a factor node
- Factors represent computation

\[
\psi(A, B, C)
\]

\[
P(A)P(B)P(C|A, B)
\]
Outline

Life without Graphical Models

Representation
  Directed Graphical Models (Bayesian Networks)
  Undirected Graphical Models (Markov Random Fields)

Inference
  Exact Inference
  Markov Chain Monte Carlo
  Variational Inference
    Loopy Belief Propagation
    Mean Field Algorithm
    Exponential Family
  Maximizing Problems

Parameter Learning

Structure Learning
Outline

Life without Graphical Models

Representation
  Directed Graphical Models (Bayesian Networks)
  Undirected Graphical Models (Markov Random Fields)

Inference
  Exact Inference
    Markov Chain Monte Carlo
  Variational Inference
    Loopy Belief Propagation
    Mean Field Algorithm
    Exponential Family
  Maximizing Problems

Parameter Learning

Structure Learning
Inference by Enumeration

- Let $X = (X_Q, X_E, X_O)$ for query, evidence, and other variables.
- Infer $P(X_Q \mid X_E)$
- By definition

$$P(X_Q \mid X_E) = \frac{P(X_Q, X_E)}{P(X_E)} = \frac{\sum_{X_O} P(X_Q, X_E, X_O)}{\sum_{X_Q, X_O} P(X_Q, X_E, X_O)}$$

- Summing exponential number of terms: with $k$ variables in $X_O$ each taking $r$ values, there are $r^k$ terms
There are a bunch of “other” variables $x_1, \ldots, x_k$

We sum over $r$ values each variable can take $\sum_{x_i=v_1}^{v_r}$

This is exponential ($r^k$): $\sum_{x_1 \ldots x_k}$

We want $\sum_{x_1 \ldots x_k} p(X)$

For a graphical model, the joint probability factors

$$p(X) = \prod_{j=1}^{m} f_j(X_{(j)})$$

Each factor $f_j$ operates on $X_{(j)} \subseteq X$
Eliminating a Variable

- Rearrange factors \( \sum_{x_1 \ldots x_k} f_1^- \ldots f_l^- f_{l+1}^+ \ldots f_m^+ \) by whether \( x_1 \in X(j) \)
- Obviously equivalent: \( \sum_{x_2 \ldots x_k} f_1^- \ldots f_l^- (\sum_{x_1} f_{l+1}^+ \ldots f_m^+) \)
- Introduce a new factor \( f_{m+1}^- = (\sum_{x_1} f_{l+1}^+ \ldots f_m^+) \)
- \( f_{m+1}^- \) contains the union of variables in \( f_{l+1}^+ \ldots f_m^+ \) except \( x_1 \)
- In fact, \( x_1 \) disappears altogether in \( \sum_{x_2 \ldots x_k} f_1^- \ldots f_l^- f_{m+1}^- \)
- Dynamic programming: compute \( f_{m+1}^- \) once, use it thereafter
- Hope: \( f_{m+1}^- \) contains very few variables
- Recursively eliminate other variables in turn
Example: Chain Graph

- Binary variables
- Say we want \( P(D) = \sum_{A,B,C} P(A)P(B|A)P(C|B)P(D|C) \)
- Let \( f_1(A) = P(A) \). Note \( f_1 \) is an array of size two:
  \[
P(A = 0) \\
P(A = 1)
\]
- \( f_2(A, B) \) is a table of size four:
  \[
P(B = 0|A = 0) \\
P(B = 0|A = 1) \\
P(B = 1|A = 0) \\
P(B = 1|A = 1)
\]
- \( \sum_{A,B,C} f_1(A)f_2(A, B)f_3(B, C)f_4(C, D) = \sum_{B,C} f_3(B, C)f_4(C, D)(\sum_A f_1(A)f_2(A, B)) \)
Example: Chain Graph

- $f_1(A) f_2(A, B)$ an array of size four: match $A$ values
  - $P(A = 0) P(B = 0 | A = 0)$
  - $P(A = 1) P(B = 0 | A = 1)$
  - $P(A = 0) P(B = 1 | A = 0)$
  - $P(A = 1) P(B = 1 | A = 1)$

- $f_5(B) \equiv \sum_A f_1(A) f_2(A, B)$ an array of size two
  - $P(A = 0) P(B = 0 | A = 0) + P(A = 1) P(B = 0 | A = 1)$
  - $P(A = 0) P(B = 1 | A = 0) + P(A = 1) P(B = 1 | A = 1)$

- For this example, $f_5(B)$ happens to be $P(B)$

- $\sum_{B,C} f_3(B, C) f_4(C, D) f_5(B) = \sum_C f_4(C, D) (\sum_B f_3(B, C) f_5(B))$, and so on

- In the end, $f_7(D) = (P(D = 0), P(D = 1))$
Example: Chain Graph

- Computation for $P(D)$: $12 \times, 6 +$
- Enumeration: $48 \times, 14 +$
- Saving depends on elimination order. Finding optimal order NP-hard; there are heuristic methods.
- Saving depends more critically on the graph structure (tree width), can be intractable
Handling Evidence

- For evidence variables $X_E$, simply plug in their value $e$
- Eliminate variables $X_O = X - X_E - X_Q$
- The final factor will be the joint $f(X_Q) = P(X_Q, X_E = e)$
- Normalize to answer query:

$$P(X_Q \mid X_E = e) = \frac{f(X_Q)}{\sum_{X_Q} f(X_Q)}$$
Summary: Exact Inference

- Enumeration
- Variable elimination
- Not covered: junction tree (aka clique tree)

Exact, but intractable for large graphs
Outline

Life without Graphical Models

Representation
   Directed Graphical Models (Bayesian Networks)
   Undirected Graphical Models (Markov Random Fields)

Inference
   Exact Inference
     Markov Chain Monte Carlo
   Variational Inference
     Loopy Belief Propagation
     Mean Field Algorithm
     Exponential Family
   Maximizing Problems

Parameter Learning

Structure Learning
Consider the inference problem \( p(X_Q = c_Q \mid X_E) \) where \( X_Q \cup X_E \subseteq \{x_1 \ldots x_n\} \)

\[
p(X_Q = c_Q \mid X_E) = \int 1(x_Q = c_Q)p(x_Q \mid X_E)dx_Q
\]

If we can draw samples \( x_Q^{(1)}, \ldots x_Q^{(m)} \sim p(x_Q \mid X_E) \), an unbiased estimator is

\[
p(X_Q = c_Q \mid X_E) \approx \frac{1}{m} \sum_{i=1}^{m} 1(x_Q^{(i)} = c_Q)
\]

The variance of the estimator decreases as \( \sqrt{V}/m \)

Inference reduces to sampling from \( p(x_Q \mid X_E) \)
To generate a sample $X = (B, E, A, J, M)$:

1. Sample $B \sim \text{Ber}(0.001)$: $r \sim U(0, 1)$. If ($r < 0.001$) then $B = 1$ else $B = 0$

2. Sample $E \sim \text{Ber}(0.002)$

3. If $B = 1$ and $E = 1$, sample $A \sim \text{Ber}(0.95)$, and so on

4. If $A = 1$ sample $J \sim \text{Ber}(0.9)$ else $J \sim \text{Ber}(0.05)$

5. If $A = 1$ sample $M \sim \text{Ber}(0.7)$ else $M \sim \text{Ber}(0.01)$

Works for Bayesian networks.
Inference with Forward Sampling

▶ Say the inference task is \( P(B = 1 \mid E = 1, M = 1) \)
▶ **Throw away** all samples except those with \( (E = 1, M = 1) \)

\[
p(B = 1 \mid E = 1, M = 1) \approx \frac{1}{m} \sum_{i=1}^{m} 1(B(i) = 1)
\]

where \( m \) is the number of surviving samples
▶ Can be highly inefficient (note \( P(E = 1) \) tiny)
▶ Does not work for Markov Random Fields
Gibbs Sampler Example: \( P(B = 1 \mid E = 1, M = 1) \)

- Gibbs sampler is a Markov Chain Monte Carlo (MCMC) method.
- Directly sample from \( p(x_Q \mid X_E) \)
- Works for both graphical models
- Initialization:
  - Fix evidence; randomly set other variables
  - e.g. \( X^{(0)} = (B = 0, E = 1, A = 0, J = 0, M = 1) \)

\[
\begin{align*}
P(A \mid B, E) &= 0.95 \\
P(A \mid B, \neg E) &= 0.94 \\
P(A \mid \neg B, E) &= 0.29 \\
P(A \mid \neg B, \neg E) &= 0.001 \\
P(B) &= 0.001 \\
P(E) &= 0.002 \\
P(J \mid A) &= 0.9 \\
P(J \mid \neg A) &= 0.05 \\
P(M \mid A) &= 0.7 \\
P(M \mid \neg A) &= 0.01 
\end{align*}
\]
Gibbs Update

- For each non-evidence variable $x_i$, fixing all other nodes $X_{-i}$, resample its value $x_i \sim P(x_i \mid X_{-i})$
- This is equivalent to $x_i \sim P(x_i \mid \text{MarkovBlanket}(x_i))$
- For a Bayesian network $\text{MarkovBlanket}(x_i)$ includes $x_i$'s parents, spouses, and children

$$P(x_i \mid \text{MarkovBlanket}(x_i)) \propto P(x_i \mid Pa(x_i)) \prod_{y \in C(x_i)} P(y \mid Pa(y))$$

where $Pa(x)$ are the parents of $x$, and $C(x)$ the children of $x$.
- For many graphical models the Markov Blanket is small.
- For example,

$$B \sim P(B \mid E = 1, A = 0) \propto P(B)P(A = 0 \mid B, E = 1)$$

$$P(A \mid B, E) = 0.95$$
$$P(A \mid B, \neg E) = 0.94$$
$$P(A \mid \neg B, E) = 0.29$$
$$P(A \mid \neg B, \neg E) = 0.001$$

$$P(B) = 0.001$$
$$P(E) = 0.002$$
Say we sampled $B = 1$. Then
$X^{(1)} = (B = 1, E = 1, A = 0, J = 0, M = 1)$

Starting from $X^{(1)}$, sample
$A \sim P(A \mid B = 1, E = 1, J = 0, M = 1)$ to get $X^{(2)}$

Move on to $J$, then repeat $B, A, J, B, A, J \ldots$

Keep all *later* samples. $P(B = 1 \mid E = 1, M = 1)$ is the fraction of samples with $B = 1$. 

\[
\begin{align*}
P(B) &= 0.001 \\
P(E) &= 0.002 \\
P(A \mid B, E) &= 0.95 \\
P(A \mid B, \neg E) &= 0.94 \\
P(A \mid \neg B, E) &= 0.29 \\
P(A \mid \neg B, \neg E) &= 0.001 \\
P(J \mid A) &= 0.9 \\
P(J \mid \neg A) &= 0.05 \\
P(M \mid A) &= 0.7 \\
P(M \mid \neg A) &= 0.01
\end{align*}
\]
Gibbs Example 2: The Ising Model

This is an undirected model with $x \in \{0, 1\}$.

$$p_{\theta}(x) = \frac{1}{Z} \exp \left( \sum_{s \in V} \theta_s x_s + \sum_{(s, t) \in E} \theta_{st} x_s x_t \right)$$
Gibbs Example 2: The Ising Model

- The Markov blanket of $x_s$ is $A, B, C, D$
- In general for undirected graphical models

\[ p(x_s \mid x_{-s}) = p(x_s \mid x_{N(s)}) \]

$N(s)$ is the neighbors of $s$.
- The Gibbs update is

\[ p(x_s = 1 \mid x_{N(s)}) = \frac{1}{\exp\left(-\left(\theta_s + \sum_{t \in N(s)} \theta_{st} x_t\right)\right) + 1} \]
Gibbs Sampling as a Markov Chain

- A Markov chain is defined by a transition matrix $T(X' \mid X)$
- Certain Markov chains have a stationary distribution $\pi$ such that $\pi = T\pi$
- Gibbs sampler is such a Markov chain with $T_i((X_{-i}, x_i') \mid (X_{-i}, x_i)) = p(x_i' \mid X_{-i})$, and stationary distribution $p(x_Q \mid X_E)$
- But it takes time for the chain to reach stationary distribution (mix)
  - Can be difficult to assert mixing
  - In practice “burn in”: discard $X^{(0)}, \ldots, X^{(T)}$
  - Use all of $X^{(T+1)}, \ldots$ for inference (they are correlated)
  - Do not thin
Collapsed Gibbs Sampling

- In general, \( \mathbb{E}_p[f(X)] \approx \frac{1}{m} \sum_{i=1}^{m} f(X^{(i)}) \) if \( X^{(i)} \sim p \)
- Sometimes \( X = (Y, Z) \) where \( Z \) has closed-form operations
- If so,

\[
\mathbb{E}_p[f(X)] = \mathbb{E}_p(Y)\mathbb{E}_p(Z|Y)[f(Y, Z)] \\
\approx \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_p(Z|Y^{(i)})[f(Y^{(i)}, Z)]
\]

if \( Y^{(i)} \sim p(Y) \)
- No need to sample \( Z \): it is collapsed
- Collapsed Gibbs sampler \( T_i((Y_{-i}, y'_i) \mid (Y_{-i}, y_i)) = p(y'_i \mid Y_{-i}) \)
- Note \( p(y'_i \mid Y_{-i}) = \int p(y'_i, Z \mid Y_{-i})dZ \)

\[
\]
Collapse $\theta$, $\phi$, Gibbs update:

$$P(z_i = j \mid z_{-i}, w) \propto \frac{n^{(w_i)} - i,j + \beta n^{(d_i)} - i,j + \alpha}{n^{(\cdot)} - i,j + W \beta n^{(d_i)} - i,. + T \alpha}$$

- $n^{(w_i)}$: number of times word $w_i$ has been assigned to topic $j$, excluding the current position
- $n^{(d_i)}$: number of times a word from document $d_i$ has been assigned to topic $j$, excluding the current position
- $n^{(\cdot)}$: number of times any word has been assigned to topic $j$, excluding the current position
- $n^{(d_i)}$: length of document $d_i$, excluding the current position
Gibbs sampling
Not covered: block Gibbs, Metropolis-Hastings

Unbiased (after burn-in), but can have high variance

To learn more, come to Prof. Prasad Tetali’s tutorial “Markov Chain Mixing with Applications” 2pm Monday.
Outline

Life without Graphical Models

Representation
  Directed Graphical Models (Bayesian Networks)
  Undirected Graphical Models (Markov Random Fields)

Inference
  Exact Inference
  Markov Chain Monte Carlo
  Variational Inference
    Loopy Belief Propagation
    Mean Field Algorithm
    Exponential Family
  Maximizing Problems

Parameter Learning

Structure Learning
Outline

Life without Graphical Models

Representation
  Directed Graphical Models (Bayesian Networks)
  Undirected Graphical Models (Markov Random Fields)

Inference
  Exact Inference
  Markov Chain Monte Carlo
  Variational Inference
    Loopy Belief Propagation
    Mean Field Algorithm
    Exponential Family
  Maximizing Problems

Parameter Learning

Structure Learning
The Sum-Product Algorithm

- Also known as belief propagation (BP)
- Exact if the graph is a tree; otherwise known as “loopy BP”, approximate
- The algorithm involves passing messages on the factor graph
- Alternative view: variational approximation (more later)
Example: A Simple HMM

- The Hidden Markov Model template (not a graphical model)

\[
\begin{align*}
\pi_1 &= \pi_2 = \frac{1}{2} \\
P(x | z=1) &= (\frac{1}{2}, \frac{1}{4}, \frac{1}{4}) \\
&\quad \text{R G B} \\
P(x | z=2) &= (\frac{1}{4}, \frac{1}{2}, \frac{1}{4}) \\
&\quad \text{R G B}
\end{align*}
\]

- Observing \( x_1 = R, x_2 = G \), the directed graphical model

- Factor graph

\[
\begin{align*}
P(z_1)P(x_1 | z_1) &\quad P(z_2 | z_1)P(x_2 | z_2)
\end{align*}
\]
Messages

A message is a vector of length $K$, where $K$ is the number of values $x$ takes.

There are two types of messages:

1. $\mu_{f \rightarrow x}$: message from a factor node $f$ to a variable node $x$
   
   $\mu_{f \rightarrow x}(i)$ is the $i$th element, $i = 1 \ldots K$.

2. $\mu_{x \rightarrow f}$: message from a variable node $x$ to a factor node $f$
Assume tree factor graph. Pick an arbitrary root, say $z_2$

Start messages at leaves.

If a leaf is a factor node $f$, $\mu_{f \rightarrow x}(x) = f(x)$

If a leaf is a variable node $x$, $\mu_{x \rightarrow f}(x) = 1$

\[
\begin{align*}
\mu_{f_1 \rightarrow z_1}(z_1 = 1) &= P(z_1 = 1)P(R | z_1 = 1) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} \\
\mu_{f_1 \rightarrow z_1}(z_1 = 2) &= P(z_1 = 2)P(R | z_1 = 2) = \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8}
\end{align*}
\]
A node (factor or variable) can send out a message if all other incoming messages have arrived.

Let $x$ be in factor $f_s$.

$\mu_{x \rightarrow f_s}(x) = \prod_{f \in \text{ne}(x) \setminus f_s} \mu_{f \rightarrow x}(x)$

$\text{ne}(x) \setminus f_s$ are factors connected to $x$ excluding $f_s$.

$\mu_{z_1 \rightarrow f_2}(z_1 = 1) = 1/4$

$\mu_{z_1 \rightarrow f_2}(z_1 = 2) = 1/8$

$P(x | z=1) = (1/2, 1/4, 1/4)$

$P(x | z=2) = (1/4, 1/2, 1/4)$

$\pi_1 = \pi_2 = 1/2$
Let $x$ be in factor $f_s$. Let the other variables in $f_s$ be $x_1:M$.

$$\mu_{f_s \rightarrow x}(x) = \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m=1}^{M} \mu_{x_m \rightarrow f_s}(x_m)$$

\[ P(z_1)P(x_1 | z_1) \quad P(z_2 | z_1)P(x_2 | z_2) \]

$$\mu_{f_2 \rightarrow z_2}(s) = \sum_{s' = 1}^{2} \mu_{z_1 \rightarrow f_2}(s') \cdot f_2(z_1 = s', z_2 = s)$$

$$= 1/4 P(z_2 = s | z_1 = 1)P(x_2 = G | z_2 = s) + 1/8 P(z_2 = s | z_1 = 2)P(x_2 = G | z_2 = s)$$

We get

$$\mu_{f_2 \rightarrow z_2}(z_2 = 1) = 1/32$$

$$\mu_{f_2 \rightarrow z_2}(z_2 = 2) = 1/8$$
The message has reached the root, pass it back down.

\[ P(z_1)P(x_1 \mid z_1) \quad P(z_2 \mid z_1)P(x_2 \mid z_2) \]

\[ \mu_{z_2 \rightarrow f_2}(z_2 = 1) = 1 \]
\[ \mu_{z_2 \rightarrow f_2}(z_2 = 2) = 1 \]

\[ P(x \mid z=1) = (1/2, 1/4, 1/4) \quad P(x \mid z=2) = (1/4, 1/2, 1/4) \]

\[ \pi_1 = \pi_2 = 1/2 \]
\[
\begin{align*}
\mu_{f_2 \rightarrow z_1}(s) &= \sum_{s' = 1}^{2} \mu_{z_2 \rightarrow f_2}(s') f_2(z_1 = s, z_2 = s') \\
&= 1 P(z_2 = 1 | z_1 = s) P(x_2 = G | z_2 = 1) \\
&\quad + 1 P(z_2 = 2 | z_1 = s) P(x_2 = G | z_2 = 2). \text{ We get}
\end{align*}
\]

\[
\begin{align*}
\mu_{f_2 \rightarrow z_1}(z_1 = 1) &= 7/16 \\
\mu_{f_2 \rightarrow z_1}(z_1 = 2) &= 3/8
\end{align*}
\]
Once a variable receives all incoming messages, we compute its marginal as

\[ p(x) \propto \prod_{f \in ne(x)} \mu_{f \rightarrow x}(x) \]

In this example

\[ P(z_1 | x_1, x_2) \propto \mu_{f_1 \rightarrow z_1} \cdot \mu_{f_2 \rightarrow z_1} = \left( \frac{1}{4} \right) \cdot \left( \frac{7}{16} \right) = \left( \frac{7}{64} \right) \Rightarrow \left( \frac{0.7}{0.3} \right) \]

\[ P(z_2 | x_1, x_2) \propto \mu_{f_2 \rightarrow z_2} = \left( \frac{1}{32} \right) \Rightarrow \left( \frac{0.2}{0.8} \right) \]

One can also compute the marginal of the set of variables \( x_s \) involved in a factor \( f_s \)

\[ p(x_s) \propto f_s(x_s) \prod_{x \in ne(f)} \mu_{x \rightarrow f}(x) \]
Handling Evidence

Observing $x = v$,

- we can absorb it in the factor (as we did); or
- set messages $\mu_{x \rightarrow f}(x) = 0$ for all $x \neq v$

Observing $X_E$,

- multiplying the incoming messages to $x \notin X_E$ gives the joint (not $p(x|X_E)$):

$$p(x, X_E) \propto \prod_{f \in \text{ne}(x)} \mu_{f \rightarrow x}(x)$$

- The conditional is easily obtained by normalization

$$p(x|X_E) = \frac{p(x, X_E)}{\sum_{x'} p(x', X_E)}$$
Loopy Belief Propagation

- So far, we assumed a tree graph
- When the factor graph contains loops, pass messages indefinitely until convergence
- But convergence may not happen
- But in many cases loopy BP still works well, empirically
Outline

Life without Graphical Models

Representation
  Directed Graphical Models (Bayesian Networks)
  Undirected Graphical Models (Markov Random Fields)

Inference
  Exact Inference
  Markov Chain Monte Carlo
  Variational Inference
    Loopy Belief Propagation
    Mean Field Algorithm
    Exponential Family
  Maximizing Problems

Parameter Learning

Structure Learning
The random variables $x$ take values in $\{0, 1\}$.

$$p_\theta(x) = \frac{1}{Z} \exp \left( \sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t \right)$$
Markovian: the conditional distribution for $x_s$ is

$$p(x_s \mid x_{-s}) = p(x_s \mid x_{N(s)})$$

$N(s)$ is the neighbors of $s$.

This reduces to

$$p(x_s = 1 \mid x_{N(s)}) = \frac{1}{\exp\left(-\left(\theta_s + \sum_{t \in N(s)} \theta_{st} x_t\right)\right) + 1}$$

Gibbs sampling would draw $x_s$ like this.
The Mean Field Algorithm for Ising Model

\[
p(x_s = 1 \mid x_{N(s)}) = \frac{1}{\exp(- (\theta_s + \sum_{t \in N(s)} \theta_{st} x_t)) + 1}
\]

- Instead of Gibbs sampling, let \( \mu_s \) be the estimated marginal \( p(x_s = 1) \)

\[
\mu_s \leftarrow \frac{1}{\exp(- (\theta_s + \sum_{t \in N(s)} \theta_{st} \mu_t)) + 1}
\]

- The \( \mu \)'s are updated iteratively

- The Mean Field algorithm is coordinate ascent and guaranteed to converge to a local optimal (more later).
Outline

Life without Graphical Models

Representation
  Directed Graphical Models (Bayesian Networks)
  Undirected Graphical Models (Markov Random Fields)

Inference
  Exact Inference
  Markov Chain Monte Carlo
  Variational Inference
    Loopy Belief Propagation
    Mean Field Algorithm
    Exponential Family
  Maximizing Problems

Parameter Learning

Structure Learning
Let $\phi(X) = (\phi_1(X), \ldots, \phi_d(X))^\top$ be $d$ sufficient statistics, where $\phi_i : \mathcal{X} \mapsto \mathbb{R}$.

Note $X$ is all the nodes in a Graphical model.

$\phi_i(X)$ sometimes called a feature function.

Let $\theta = (\theta_1, \ldots, \theta_d)^\top \in \mathbb{R}^d$ be canonical parameters.

The exponential family is a family of probability densities:

$$p_\theta(x) = \exp \left( \theta^\top \phi(x) - A(\theta) \right)$$
Exponential Family

\[ p_\theta(x) = \exp \left( \theta^\top \phi(x) - A(\theta) \right) \]

- The key is the inner product between parameters \( \theta \) and sufficient statistics \( \phi \).
- \( A \) is the log partition function,
  \[ A(\theta) = \log \int \exp \left( \theta^\top \phi(x) \right) \nu(dx) \]
- \( A = \log Z \)
Parameters for which the density is normalizable:

$$\Omega = \{ \theta \in \mathbb{R}^d \mid A(\theta) < \infty \}$$

A *minimal* exponential family is where the $\phi$’s are linearly independent.

An *overcomplete* exponential family is where the $\phi$’s are linearly dependent:

$$\exists \alpha \in \mathbb{R}^d, \alpha^\top \phi(x) = \text{constant} \ \forall x$$

Both minimal and overcomplete representations are useful.
Exponential Family Example 1: Bernoulli

\[ p(x) = \beta^x (1 - \beta)^{1-x} \text{ for } x \in \{0, 1\} \text{ and } \beta \in (0, 1). \]

▶ Does not look like an exponential family!
▶ Can be rewritten as

\[ p(x) = \exp(x \log \beta + (1 - x) \log(1 - \beta)) \]

▶ Now in exponential family form with
\[ \phi_1(x) = x, \phi_2(x) = 1 - x, \theta_1 = \log \beta, \theta_2 = \log(1 - \beta), \text{ and } A(\theta) = 0. \]
▶ Overcomplete: \( \alpha_1 = \alpha_2 = 1 \) makes \( \alpha^\top \phi(x) = 1 \) for all \( x \).
Exponential Family Example 1: Bernoulli

\[ p(x) = \exp (x \log \beta + (1 - x) \log (1 - \beta)) \]

Can be further rewritten as

\[ p(x) = \exp (x \theta - \log (1 + \exp(\theta))) \]

Minimal exponential family with

\[ \phi(x) = x, \theta = \log \frac{\beta}{1-\beta}, A(\theta) = \log (1 + \exp(\theta)). \]

Many distributions (e.g., Gaussian, exponential, Poisson, Beta) are in the exponential family, but not all (e.g., the Laplace distribution).
**Exponential Family Example 2: Ising Model**

\[
p_\theta(x) = \exp \left( \sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t - A(\theta) \right)
\]

- Binary random variable \( x_s \in \{0, 1\} \)
- \( d = |V| + |E| \) sufficient statistics: \( \phi(x) = (\ldots x_s \ldots x_{st} \ldots)^\top \)
- This is a regular (\( \Omega = \mathbb{R}^d \)), minimal exponential family.
Similar to Ising model but generalizing $x_s \in \{0, \ldots, r - 1\}$.

Indicator functions $f_{sj}(x) = 1$ if $x_s = j$ and 0 otherwise, and $f_{stjk}(x) = 1$ if $x_s = j \land x_t = k$, and 0 otherwise.

$$p_\theta(x) = \exp \left( \sum_{sj} \theta_{sj} f_{sj}(x) + \sum_{stjk} \theta_{stjk} f_{stjk}(x) - A(\theta) \right)$$

$d = r|V| + r^2|E|$

Regular but overcomplete, because $\sum_{j=0}^{r-1} \theta_{sj}(x) = 1$ for any $s \in V$ and all $x$.

The Potts model is a special case where the parameters are tied: $\theta_{stkk} = \alpha$, and $\theta_{stjk} = \beta$ for $j \neq k$. 
For sufficient statistics defined by indicator functions

- e.g., $\phi_{sj}(x) = f_{sj}(x) = 1$ if $x_s = j$ and 0 otherwise
- The marginal can be obtained via the mean

$$\mathbb{E}_\theta[\phi_{sj}(x)] = P(x_s = j)$$

- Since inference is about computing the marginal, in this case it is equivalent to computing the mean.
Mean Parameters

- Let $p$ be any density (not necessarily in exponential family).
- Given sufficient statistics $\phi$, the mean parameters $\mu = (\mu_1, \ldots, \mu_d)^\top$ is

$$
\mu_i = \mathbb{E}_p[\phi_i(x)] = \int \phi_i(x)p(x)dx
$$

- The set of mean parameters

$$
\mathcal{M} = \{\mu \in \mathbb{R}^d \mid \exists p \text{ s.t. } \mathbb{E}_p[\phi(x)] = \mu\}
$$

- If $\mu^{(1)}, \mu^{(2)} \in \mathcal{M}$, there must exist $p^{(1)}, p^{(2)}$
- The convex combinations of $p^{(1)}, p^{(2)}$ leads to another mean parameter in $\mathcal{M}$
- Therefore $\mathcal{M}$ is convex
Example: The First Two Moments

- Let $\phi_1(x) = x, \phi_2(x) = x^2$
- For any $p$ (not necessarily Gaussian) on $x$, the mean parameters $\mu = (\mu_1, \mu_2) = (\mathbb{E}(x), \mathbb{E}(x^2))^\top$.
- Note $\mathbb{V}(x) = \mathbb{E}(x^2) - \mathbb{E}^2(x) = \mu_2 - \mu_1^2 \geq 0$ for any $p$
- $\mathcal{M}$ is not $\mathbb{R}^2$ but rather the subset $\mu_1 \in \mathbb{R}, \mu_2 \geq \mu_1^2$. 
The marginal polytope is defined for discrete $x_s$.

Recall $\mathcal{M} = \{\mu \in \mathbb{R}^d \mid \mu = \sum_x \phi(x)p(x) \text{ for some } p\}$

$p$ can be a point mass function on a particular $x$.

In fact any $p$ is a convex combination of such point mass functions.

$\mathcal{M} = \text{conv}\{\phi(x), \forall x\}$ is a convex hull, called the marginal polytope.
Tiny Ising model: two nodes $x_1, x_2 \in \{0, 1\}$ connected by an edge.

- minimal sufficient statistics $\phi(x_1, x_2) = (x_1, x_2, x_1x_2)^\top$.
- only 4 different $x = (x_1, x_2)$.
- the marginal polytope is $\mathcal{M} = \text{conv}\{(0, 0, 0), (0, 1, 0), (1, 0, 0), (1, 1, 1)\}$

- the convex hull is a polytope inside the unit cube.
- the three coordinates are node marginals $\mu_1 \equiv \mathbb{E}_p[x_1 = 1]$, $\mu_2 \equiv \mathbb{E}_p[x_2 = 1]$ and edge marginal $\mu_{12} \equiv \mathbb{E}_p[x_1 = x_2 = 1]$, hence the name.
For any regular exponential family, $A(\theta)$ is convex in $\theta$.

Strictly convex for minimal exponential family.

Nice property: \[
\frac{\partial A(\theta)}{\partial \theta_i} = \mathbb{E}_\theta[\phi_i(x)]
\]

Therefore, $\nabla A = \mu$, the mean parameters of $p_\theta$. 
The **conjugate dual function** $A^*$ to $A$ is defined as

$$A^*(\mu) = \sup_{\theta \in \Omega} \theta^\top \mu - A(\theta)$$

Such definition, where a quantity is expressed as the solution to an optimization problem, is called a *variational* definition.

- For any $\mu \in \mathcal{M}$’s interior, let $\theta(\mu)$ satisfy
  $$\mathbb{E}_{\theta(\mu)}[\phi(x)] = \nabla A(\theta(\mu)) = \mu.$$

- Then $A^*(\mu) = -H(p_{\theta(\mu)})$ the negative entropy.

- The dual of the dual gives back $A$:
  $$A(\theta) = \sup_{\mu \in \mathcal{M}} \mu^\top \theta - A^*(\mu)$$

- For all $\theta \in \Omega$, the supremum is attained uniquely at the $\mu \in \mathcal{M}^0$ by the moment matching conditions $\mu = \mathbb{E}_\theta[\phi(x)]$. 
Example: Conjugate Dual for Bernoulli

Recall the minimal exponential family for Bernoulli with \( \phi(x) = x, A(\theta) = \log(1 + \exp(\theta)), \Omega = \mathbb{R}. \)

By definition

\[
A^*(\mu) = \sup_{\theta \in \mathbb{R}} \theta \mu - \log(1 + \exp(\theta))
\]

Taking derivative and solve

\[
A^*(\mu) = \mu \log \mu + (1 - \mu) \log(1 - \mu)
\]

i.e., the negative entropy.
Inference with Variational Representation

\[ A(\theta) = \sup_{\mu \in \mathcal{M}} \mu^\top \theta - A^*(\mu) \] is attained by \( \mu = \mathbb{E}_\theta[\phi(x)] \).

- Want to compute the marginals \( P(x_s = j) \)? They are the mean parameters \( \mu_{sj} = \mathbb{E}_\theta[\phi_{ij}(x)] \) under standard overcomplete representation.

- Want to compute the mean parameters \( \mu_{sj} \)? They are the \( \text{arg sup} \) to the optimization problem above.

- This variational representation is exact, not approximate (will relax it next to derive loopy BP and mean field).
The Difficulties with Variational Representation

\[ A(\theta) = \sup_{\mu \in \mathcal{M}} \mu^\top \theta - A^*(\mu) \]

- Difficult to solve even though it is a convex problem
- Two issues:
  - Although the marginal polytope \( \mathcal{M} \) is convex, it can be quite complex (exponential number of vertices)
  - The dual function \( A^*(\mu) \) usually does not admit an explicit form.
- Variational approximation modifies the optimization problem so that it is tractable, at the price of an approximate solution.
- Next, we cast mean field and sum-product algorithms as variational approximations.
The mean field method replaces \( \mathcal{M} \) with a simpler subset \( \mathcal{M}(F) \) on which \( A^*(\mu) \) has a closed form.

Consider the fully disconnected subgraph \( F = (V, \emptyset) \) of the original graph \( G = (V, E) \)

Set all \( \theta_i = 0 \) if \( \phi_i \) involves edges

The densities in this sub-family are all fully factorized:

\[
p_\theta(x) = \prod_{s \in V} p(x_s; \theta_s)
\]
The Geometry of $\mathcal{M}(F)$

- Let $\mathcal{M}(F)$ be the mean parameters of the fully factorized sub-family. In general, $\mathcal{M}(F) \subset \mathcal{M}$
- Recall $\mathcal{M}$ is the convex hull of extreme points $\{\phi(x)\}$.
- It turns out the extreme points $\{\phi(x)\} \in \mathcal{M}(F)$.
- Example:
  - The tiny Ising model $x_1, x_2 \in \{0, 1\}$ with $\phi = (x_1, x_2, x_1 x_2)\top$
  - The point mass distribution $p(x = (0, 1)\top) = 1$ is realized as a limit to the series $p(x) = \exp(\theta_1 x_1 + \theta_2 x_2 - A(\theta))$ where $\theta_1 \to -\infty$ and $\theta_2 \to \infty$.
  - This series is in $F$ because $\theta_{12} = 0$.
  - Hence the extreme point $\phi(x) = (0, 1, 0)$ is in $\mathcal{M}(F)$. 

![Diagram of the Geometry of $\mathcal{M}(F)$]
Because the extreme points of $\mathcal{M}$ are in $\mathcal{M}(F)$, if $\mathcal{M}(F)$ were convex, we would have $\mathcal{M} = \mathcal{M}(F)$.

But in general $\mathcal{M}(F)$ is a true subset of $\mathcal{M}$.

Therefore, $\mathcal{M}(F)$ is a nonconvex inner set of $\mathcal{M}$.
The Mean Field Method as Variational Approximation

- Recall the exact variational problem

\[ A(\theta) = \sup_{\mu \in \mathcal{M}} \mu^\top \theta - A^*(\mu) \]

attained by solution to inference problem \( \mu = \mathbb{E}_\theta[\phi(x)] \).

- The mean field method simply replaces \( \mathcal{M} \) with \( \mathcal{M}(F) \)

\[ \mathcal{L}(\theta) = \sup_{\mu \in \mathcal{M}(F)} \mu^\top \theta - A^*(\mu) \]

- Obvious \( \mathcal{L}(\theta) \leq A(\theta) \).

- The original solution \( \mu^* \) may not be in \( \mathcal{M}(F) \)

- Even if \( \mu^* \in \mathcal{M}(F) \), may hit local maximum and not find it

- Why both? Because \( A^*(\mu) = -H(p_{\theta}(\mu)) \) has a very simple form for \( \mathcal{M}(F) \)
Example: Mean Field for Ising Model

- The mean parameters for the Ising model are the node and edge marginals: \( \mu_s = p(x_x = 1) \), \( \mu_{st} = p(x_s = 1, x_t = 1) \)
- Fully factorized \( \mathcal{M}(F') \) means no edge. \( \mu_{st} = \mu_s \mu_t \)
- For \( \mathcal{M}(F') \), the dual function \( A^*(\mu) \) has the simple form

\[
A^*(\mu) = \sum_{s \in V} -H(\mu_s) = \sum_{s \in V} \mu_s \log \mu_s + (1 - \mu_s) \log (1 - \mu_s)
\]

- Thus the mean field problem is

\[
\mathcal{L}(\theta) = \sup_{\mu \in \mathcal{M}(F)} \mu^\top \theta - \sum_{s \in V} (\mu_s \log \mu_s + (1 - \mu_s) \log (1 - \mu_s))
\]

\[
= \max_{(\mu_1 \ldots \mu_m) \in [0,1]^m} \left( \sum_{s \in V} \theta_s \mu_s + \sum_{(s,t) \in E} \theta_{st} \mu_s \mu_t + \sum_{s \in V} H(\mu_s) \right)
\]
Example: Mean Field for Ising Model

$$\mathcal{L}(\theta) = \max_{(\mu_1...\mu_m) \in [0,1]^m} \left( \sum_{s \in V} \theta_s \mu_s + \sum_{(s,t) \in E} \theta_{st} \mu_s \mu_t + \sum_{s \in V} H(\mu_s) \right)$$

- Bilinear in $\mu$, not jointly concave
- But concave in a single dimension $\mu_s$, fixing others.
- Iterative coordinate-wise maximization: fixing $\mu_t$ for $t \neq s$ and optimizing $\mu_s$.
- Setting the partial derivative w.r.t. $\mu_s$ to 0 yields:

$$\mu_s = \frac{1}{1 + \exp \left( -\left( \theta_s + \sum_{(s,t) \in E} \theta_{st} \mu_t \right) \right)}$$

as we’ve seen before.
- Caution: mean field converges to a local maximum depending on the initialization of $\mu_1 \ldots \mu_m$. 
The sum-product algorithm makes two approximations:

- it relaxes $\mathcal{M}$ to an outer set $L$
- it replaces the dual $A^*$ with an approximation.

\[
A(\theta) = \sup_{\mu \in L} \mu^\top \theta - \tilde{A}^*(\mu)
\]
The Outer Relaxation

- For overcomplete exponential families on discrete nodes, the mean parameters are node and edge marginals
  \[ \mu_{sj} = p(x_s = j), \mu_{stjk} = p(x_s = j, x_t = k). \]
- The marginal polytope is \( \mathcal{M} = \{ \mu \mid \exists p \text{ with marginals } \mu \} \).
- Now consider \( \tau \in \mathbb{R}^d_+ \) satisfying “node normalization” and “edge-node marginal consistency” conditions:

\[
\begin{align*}
\sum_{j=0}^{r-1} \tau_{sj} &= 1 \quad \forall s \in V \\
\sum_{k=0}^{r-1} \tau_{stjk} &= \tau_{sj} \quad \forall s, t \in V, j = 0 \ldots r - 1 \\
\sum_{j=0}^{r-1} \tau_{stjk} &= \tau_{tk} \quad \forall s, t \in V, k = 0 \ldots r - 1
\end{align*}
\]
- Define \( L = \{ \tau \text{ satisfying the above conditions} \} \).
The Outer Relaxation

- If the graph is a tree then $\mathcal{M} = L$
- If the graph has cycles then $\mathcal{M} \subset L$
  - $L$ is too lax to satisfy other constraints that true marginals need to satisfy
- Nice property: $L$ is still a polytope, but much simpler than $\mathcal{M}$.

The first approximation in sum-product is to replace $\mathcal{M}$ with $L$. 
Recall $\mu$ are node and edge marginals

If the graph is a tree, one can exactly reconstruct the joint probability

$$p_{\mu}(x) = \prod_{s \in V} \mu_{sx} \prod_{(s,t) \in E} \frac{\mu_{stx}x_s}{\mu_{sx} \mu_{tx}}$$

If the graph is a tree, the entropy of the joint distribution is

$$H(p_{\mu}) = - \sum_{s \in V} \sum_{j=0}^{r-1} \mu_{sj} \log \mu_{sj} - \sum_{(s,t) \in E} \sum_{j,k} \mu_{stjk} \log \frac{\mu_{stjk}}{\mu_{sj} \mu_{tk}}$$

Neither holds for graph with cycles.
Define the *Bethe entropy* for \( \tau \in L \) on loopy graphs in the same way:

\[
H_{\text{Bethe}}(p_\tau) = -\sum_{s \in V} \sum_{j=0}^{r-1} \tau_{sj} \log \tau_{sj} - \sum_{(s,t) \in E} \sum_{j,k} \tau_{stjk} \log \frac{\tau_{stjk}}{\tau_{sj}\tau_{tk}}
\]

Note \( H_{\text{Bethe}} \) is not a true entropy. The second approximation in sum-product is to replace \( A^*(\tau) \) with \( -H_{\text{Bethe}}(p_\tau) \).
With these two approximations, we arrive at the variational problem
\[ \tilde{A}(\theta) = \sup_{\tau \in L} \tau^\top \theta + H_{\text{Bethe}}(p_\tau) \]

- Optimality conditions require the gradients vanish w.r.t. both \( \tau \) and the Lagrangian multipliers on constraints \( \tau \in L \).
- The sum-product algorithm can be derived as an iterative fixed point procedure to satisfy optimality conditions.
- At the solution, \( \tilde{A}(\theta) \) is not guaranteed to be either an upper or a lower bound of \( A(\theta) \).
- \( \tau \) may not correspond to a true marginal distribution.
Summary: Variational Inference

- The sum-product algorithm (loopy belief propagation)
- The mean field method
- Not covered: Expectation Propagation

Efficient computation. But often *unknown* bias in solution.
Outline

Life without Graphical Models

Representation
- Directed Graphical Models (Bayesian Networks)
- Undirected Graphical Models (Markov Random Fields)

Inference
- Exact Inference
- Markov Chain Monte Carlo
- Variational Inference
  - Loopy Belief Propagation
  - Mean Field Algorithm
  - Exponential Family

  Maximizing Problems

Parameter Learning

Structure Learning
Maximizing Problems

Recall the HMM example

There are two senses of “best states” $z_{1:N}$ given $x_{1:N}$:

1. So far we computed the marginal $p(z_n|x_{1:N})$
   - We can define “best” as $z_n^* = \arg\max_k p(z_n = k|x_{1:N})$
   - However $z_{1:N}^*$ as a whole may not be the best
   - In fact $z_{1:N}^*$ can even have zero probability!

2. An alternative is to find

$$z_{1:N}^* = \arg\max_{z_{1:N}} p(z_{1:N}|x_{1:N})$$

- finds the most likely state configuration as a whole
- The max-sum algorithm solves this
- Generalizes the Viterbi algorithm for HMMs
Intermediate: The Max-Product Algorithm

Simple modification to the sum-product algorithm: replace \( \sum \) with \( \max \) in the factor-to-variable messages.

\[
\mu_{f_s \rightarrow x}(x) = \max_{x_1} \ldots \max_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m=1}^{M} \mu_{x_m \rightarrow f_s}(x_m)
\]

\[
\mu_{x_m \rightarrow f_s}(x_m) = \prod_{f \in \text{ne}(x_m) \setminus f_s} \mu_{f \rightarrow x_m}(x_m)
\]

\[
\mu_{x_{\text{leaf}} \rightarrow f}(x) = 1
\]

\[
\mu_{f_{\text{leaf}} \rightarrow x}(x) = f(x)
\]
Intermediate: The Max-Product Algorithm

- As in sum-product, pick an arbitrary variable node $x$ as the root.
- Pass messages up from leaves until they reach the root.
- Unlike sum-product, do not pass messages back from root to leaves.
- At the root, multiply incoming messages

$$p_{\text{max}} = \max_x \left( \prod_{f \in \text{ne}(x)} \mu_{f \rightarrow x}(x) \right)$$

- This is the probability of the most likely state configuration.
To identify the configuration itself, keep back pointers:

When creating the message

\[
\mu_{f_s \rightarrow x}(x) = \max_{x_1} \ldots \max_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m=1}^{M} \mu_{x_m \rightarrow f_s}(x_m)
\]

for each \(x\) value, we separately create \(M\) pointers back to the values of \(x_1, \ldots, x_M\) that achieve the maximum.

At the root, backtrack the pointers.
Message from leaf $f_1$

$\mu_{f_1 \rightarrow z_1}(z_1 = 1) = P(z_1 = 1)P(R|z_1 = 1) = 1/2 \cdot 1/2 = 1/4$

$\mu_{f_1 \rightarrow z_1}(z_1 = 2) = P(z_1 = 2)P(R|z_1 = 2) = 1/2 \cdot 1/4 = 1/8$

The second message

$\mu_{z_1 \rightarrow f_2}(z_1 = 1) = 1/4$

$\mu_{z_1 \rightarrow f_2}(z_1 = 2) = 1/8$
Intermediate: The Max-Product Algorithm

\[
\mu_{f_2 \rightarrow z_2}(z_2 = 1) = \max_{z_1} f_2(z_1, z_2) \mu_{z_1 \rightarrow f_2}(z_1)
\]

\[
= \max_{z_1} P(z_2 = 1 \mid z_1) P(x_2 = G \mid z_2 = 1) \mu_{z_1 \rightarrow f_2}(z_1)
\]

\[
= \max(1/4 \cdot 1/4 \cdot 1/4, 1/2 \cdot 1/4 \cdot 1/8) = 1/64
\]

Back pointer for \(z_2 = 1\): either \(z_1 = 1\) or \(z_1 = 2\)
Intermediate: The Max-Product Algorithm

The other element of the same message:

\[
\mu_{f_2 \rightarrow z_2}(z_2 = 2) \\
= \max_{z_1} f_2(z_1, z_2) \mu_{z_1 \rightarrow f_2}(z_1) \\
= \max_{z_1} P(z_2 = 2 \mid z_1) P(x_2 = G \mid z_2 = 2) \mu_{z_1 \rightarrow f_2}(z_1) \\
= \max (3/4 \cdot 1/2 \cdot 1/4, 1/2 \cdot 1/2 \cdot 1/8) = 3/32
\]

Back pointer for \( z_2 = 2 \): \( z_1 = 1 \)
Intermediate: The Max-Product Algorithm

At root $z_2$,

\[ \max_{s=1,2} \mu_{f_2 \rightarrow z_2}(s) = \frac{3}{32} \]

\[ z_2 = 2 \rightarrow z_1 = 1 \]

\[ z_{1:2}^* = \arg \max_{z_{1:2}} p(z_{1:2}|x_{1:2}) = (1, 2) \]

In this example, sum-product and max-product produce the same best sequence; in general they differ.
The max-sum algorithm is equivalent to the max-product algorithm, but work in log space to avoid underflow.

\[
\mu_{f_s \rightarrow x}(x) = \max_{x_1 \ldots x_M} \log f_s(x, x_1, \ldots, x_M) + \sum_{m=1}^{M} \mu_{x_m \rightarrow f_s}(x_m)
\]

\[
\mu_{x_m \rightarrow f_s}(x_m) = \sum_{f \in \text{ne}(x_m) \setminus f_s} \mu_{f \rightarrow x_m}(x_m)
\]

\[
\mu_{x_{\text{leaf}} \rightarrow f}(x) = 0
\]

\[
\mu_{f_{\text{leaf}} \rightarrow x}(x) = \log f(x)
\]

When at the root,

\[
\log p^\text{max} = \max_x \left( \sum_{f \in \text{ne}(x)} \mu_{f \rightarrow x}(x) \right)
\]

The back pointers are the same.
Outline

Life without Graphical Models

Representation
  Directed Graphical Models (Bayesian Networks)
  Undirected Graphical Models (Markov Random Fields)

Inference
  Exact Inference
  Markov Chain Monte Carlo
  Variational Inference
    Loopy Belief Propagation
    Mean Field Algorithm
    Exponential Family
  Maximizing Problems

Parameter Learning

Structure Learning
Parameter Learning

- Assume the graph structure is given
- Learning in exponential family: estimate $\theta$ from iid data $x_1 \ldots x_n$.
- Principle: maximum likelihood
- Distinguish two cases:
  - fully observed data: all dimensions of $x$ are observed
  - partially observed data: some dimensions of $x$ are unobserved.
\[ p_\theta(x) = \exp(\theta^\top \phi(x) - A(\theta)) \]

- Given iid data \(x_1 \ldots x_n\), the log likelihood is

\[
\ell(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i) = \theta^\top \left( \frac{1}{n} \sum_{i=1}^{n} \phi(x_i) \right) - A(\theta) = \theta^\top \hat{\mu} - A(\theta)
\]

- \(\hat{\mu} \equiv \frac{1}{n} \sum_{i=1}^{n} \phi(x_i)\) is the mean parameter of the empirical distribution on \(x_1 \ldots x_n\). Clearly \(\hat{\mu} \in \mathcal{M}\).

- Maximum likelihood: \(\theta^{ML} = \arg \sup_{\theta \in \Omega} \theta^\top \hat{\mu} - A(\theta)\)

- The solution is \(\theta^{ML} = \theta(\hat{\mu})\), the exponential family density whose mean parameter matches \(\hat{\mu}\).

- When \(\hat{\mu} \in \mathcal{M}^0\) and \(\phi\) minimal, there is a unique maximum likelihood solution \(\theta^{ML}\).
Each item \((x, z)\) where \(x\) observed, \(z\) unobserved

Full data \((x_1, z_1) \ldots (x_n, z_n)\), but we only observe \(x_1 \ldots x_n\)

The incomplete likelihood \(\ell(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_{\theta}(x_i)\) where
\[ p_{\theta}(x_i) = \int p_{\theta}(x_i, z)dz \]

Can be written as \(\ell(\theta) = \frac{1}{n} \sum_{i=1}^{n} A_{x_i}(\theta) - A(\theta)\)

New log partition function of \(p_{\theta}(z | x_i)\), one per item:

\[ A_{x_i}(\theta) = \log \int \exp(\theta^\top \phi(x_i, z'))d\mathbf{z}' \]

Expectation-Maximization (EM) algorithm: lower bound \(A_{x_i}\)
EM as Variational Lower Bound

- Mean parameter realizable by any distribution on $z$ while holding $x_i$ fixed:
  $$\mathcal{M}_{x_i} = \{\mu \in \mathbb{R}^d \mid \mu = \mathbb{E}_p[\phi(x_i, z)] \text{ for some } p\}$$

- The variational definition $A_{x_i}(\theta) = \sup_{\mu \in \mathcal{M}_{x_i}} \theta^\top \mu - A^*_{x_i}(\mu)$

- Trivial variational lower bound:
  $$A_{x_i}(\theta) \geq \theta^\top \mu^i - A^*_{x_i}(\mu^i), \forall \mu^i \in \mathcal{M}_{x_i}$$

- Lower bound $\mathcal{L}$ on the incomplete log likelihood:

\[
\ell(\theta) = \frac{1}{n} \sum_{i=1}^{n} A_{x_i}(\theta) - A(\theta) \\
\geq \frac{1}{n} \sum_{i=1}^{n} \left(\theta^\top \mu^i - A^*_{x_i}(\mu^i)\right) - A(\theta) \\
\equiv \mathcal{L}(\mu^1, \ldots, \mu^n, \theta)
\]
Exact EM: The E-Step

The EM algorithm is coordinate ascent on $\mathcal{L}(\mu^1, \ldots, \mu^n, \theta)$.

- In the E-step, maximizes each $\mu^i$

$$
\mu^i \leftarrow \arg \max_{\mu^i \in \mathcal{M}_{x_i}} \mathcal{L}(\mu^1, \ldots, \mu^n, \theta)
$$

- Equivalently, $\arg\max_{\mu^i \in \mathcal{M}_{x_i}} \theta^\top \mu^i - A^*_{x_i}(\mu^i)$

- This is the variational representation of the mean parameter $\mu^i(\theta) = \mathbb{E}_\theta[\phi(x_i, z)]$

- The E-step is named after this $\mathbb{E}_\theta[\cdot]$ under the current parameters $\theta$
In the M-step, maximize $\theta$ holding the $\mu$’s fixed:

$$
\theta \leftarrow \arg \max_{\theta \in \Omega} \mathcal{L}(\mu^1, \ldots, \mu^n, \theta) = \arg \max_{\theta \in \Omega} \theta^\top \hat{\mu} - A(\theta)
$$

$$
\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mu^i
$$

The solution $\theta(\hat{\mu})$ satisfies $\mathbb{E}_{\theta(\hat{\mu})}[\phi(x)] = \hat{\mu}$

Standard fully observed maximum likelihood problem, hence the name M-step
Variational EM

For loopy graphs E-step often intractable.

- Can’t maximize

\[
\max_{\mu^i \in \mathcal{M}_{x_i}} \theta^\top \mu^i - A^*_x(\mu^i)
\]

- Improve but not necessarily maximize: “generalized EM”

- The mean field method maximizes

\[
\max_{\mu^i \in \mathcal{M}_{x_i}(F)} \theta^\top \mu^i - A^*_x(\mu^i)
\]

  - up to local maximum
  - recall $\mathcal{M}_{x_i}(F)$ is an inner approximation to $\mathcal{M}_{x_i}$

- Mean field E-step leads to generalized EM

- The sum-product algorithm does not lead to generalized EM
Outline

Life without Graphical Models

Representation
- Directed Graphical Models (Bayesian Networks)
- Undirected Graphical Models (Markov Random Fields)

Inference
- Exact Inference
- Markov Chain Monte Carlo
- Variational Inference
  - Loopy Belief Propagation
  - Mean Field Algorithm
  - Exponential Family
- Maximizing Problems

Parameter Learning

Structure Learning
Let $\mathcal{M}$ be all allowed candidate features

Let $M \subseteq \mathcal{M}$ be a log-linear model structure

$$P(X \mid M, \theta) = \frac{1}{Z} \exp \left( \sum_{i \in M} \theta_i f_i(X) \right)$$

A score for the model $M$ can be $\max_\theta \ln P(\text{Data} \mid M, \theta)$

The score is always better for larger $M$ – needs regularization

$M$ and $\theta$ treated separately
Consider a $p$-dimensional multivariate Gaussian $N(\mu, \Sigma)$

The graphical model has $p$ nodes $x_1, \ldots, x_p$

The edge between $x_i, x_j$ is absent if and only if $\Omega_{ij} = 0$, where $\Omega = \Sigma^{-1}$

Equivalently, $x_i, x_j$ are conditionally independent given other variables

![Diagram of a graphical model with nodes $x_1, x_2, x_3, x_4$ and edges.]
Let data be $X^{(1)}, \ldots, X^{(n)} \sim N(\mu, \Sigma)$

The log likelihood is
\[
\frac{n}{2} \log |\Omega| - \frac{1}{2} \sum_{i=1}^{n} (X^{(i)} - \mu) \top \Omega (X^{(i)} - \mu)
\]

The maximum likelihood estimate of $\Sigma$ is the sample covariance

\[
S = \frac{1}{n} \sum_{i} (X^{(i)} - \bar{X}) \top (X^{(i)} - \bar{X})
\]

where $\bar{X}$ is the sample mean

$S^{-1}$ is not a good estimate of $\Omega$ when $n$ is small
For centered data, minimize a regularized problem instead:

\[- \log |\Omega| + \frac{1}{n} \sum_{i=1}^{n} X^{(i)\top} \Omega X^{(i)} + \lambda \sum_{i \neq j} |\Omega_{ij}|\]

Known as glasso
Recap

- Given GM = joint distribution $p(x_1, \ldots, x_n)$
  - BN or MRF
  - conditional independence
- Do inference = $p(X_Q \mid X_E)$, in general
  $X_Q \cup X_E \subset \{x_1 \ldots x_n\}$
  - exact, MCMC, variational
- If $p(x_1, \ldots, x_n)$ not given, estimate it from data
  - parameter and structure learning

Much on-going research!