

CS 3750 Machine Learning
Lecture 5

Markov Random Fields III

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

CS 3750 Advanced Machine Learning

Markov random fields

- **MRF topics covered so far:**
 - MRF definition and graphical representation
 - Variable elimination
 - Tree decomposition of MRFs
 - Conversion of BBNs to MRFs and Clique trees
 - **Variable elimination** on Clique trees
 - **Belief propagation** on Clique trees
- **Today:**
 - Variable elimination on Factor graphs
 - Learning of MRFs

Markov random fields

- Probabilistic models with symmetric dependences.

- Typically models spatially varying quantities

$$P(x) \propto \prod_{c \in cl(x)} \phi_c(x_c)$$

$\phi_c(x_c)$ - A potential function (defined over factors)

- If $\phi_c(x_c)$ is strictly positive we can rewrite the definition in terms of a log-linear model :

$$P(x) = \frac{1}{Z} \exp \left(- \sum_{c \in cl(x)} E_c(x_c) \right) \quad \text{- Energy function}$$

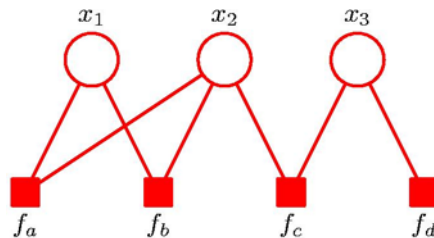
- Gibbs (Boltzman) distribution

$$Z = \sum_{x \in \{x\}} \exp \left(- \sum_{c \in cl(x)} E_c(x_c) \right) \quad \text{- A partition function}$$

CS 3750 Advanced Machine Learning

Factor Graphs

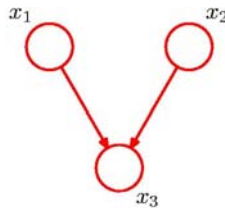
- Explicit representation of factors
 - 2 types of nodes: factors and variables



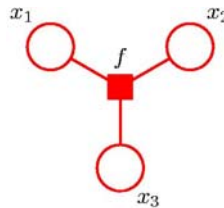
$$p(\mathbf{x}) = f_a(x_1, x_2) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3)$$

$$p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s)$$

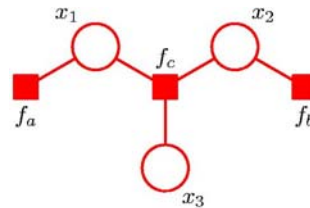
Factor Graphs from Directed Graphs



$$p(\mathbf{x}) = p(x_1)p(x_2) \\ p(x_3|x_1, x_2)$$



$$f(x_1, x_2, x_3) = \\ p(x_1)p(x_2)p(x_3|x_1, x_2)$$

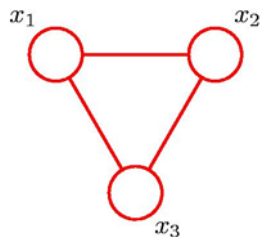


$$f_a(x_1) = p(x_1)$$

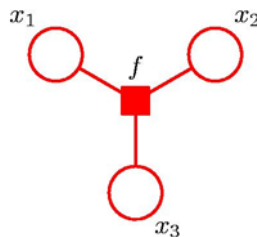
$$f_b(x_2) = p(x_2)$$

$$f_c(x_1, x_2, x_3) = p(x_3|x_1, x_2)$$

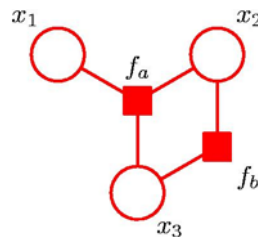
Factor Graphs from Undirected Graphs



$$\psi(x_1, x_2, x_3)$$



$$f(x_1, x_2, x_3) \\ = \psi(x_1, x_2, x_3)$$



$$f_a(x_1, x_2, x_3)f_b(x_2, x_3) \\ = \psi(x_1, x_2, x_3)$$

The Sum-Product Algorithm (1)

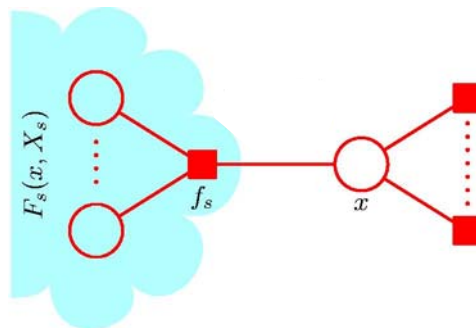
Objective:

- i. to obtain an efficient, exact inference algorithm for finding marginals;
- ii. in situations where several marginals are required, to allow computations to be shared efficiently.

Key idea: Distributive Law

$$ab + ac = a(b + c)$$

The Sum-Product Algorithm (2)



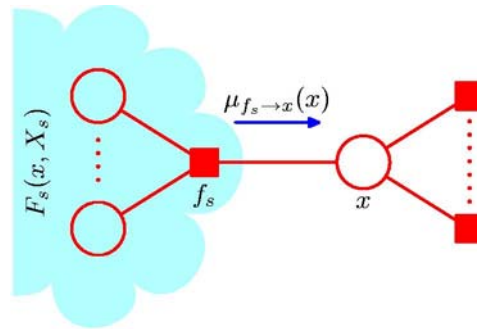
$$p(x) = \sum_{\mathbf{x} \setminus x} p(\mathbf{x})$$

$$p(\mathbf{x}) = \prod_{s \in \text{ne}(x)} F_s(x, X_s)$$

Factors that are neighbors of x

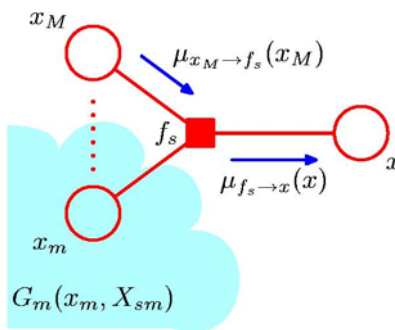
A blue arrow points from the text "Factors that are neighbors of x " to the product symbol in the equation above.

The Sum-Product Algorithm (3)



$$\begin{aligned}
 p(x) &= \prod_{s \in \text{ne}(x)} \left[\sum_{X_s} F_s(x, X_s) \right] \\
 &= \prod_{s \in \text{ne}(x)} \mu_{f_s \rightarrow x}(x). \qquad \mu_{f_s \rightarrow x}(x) \equiv \sum_{X_s} F_s(x, X_s)
 \end{aligned}$$

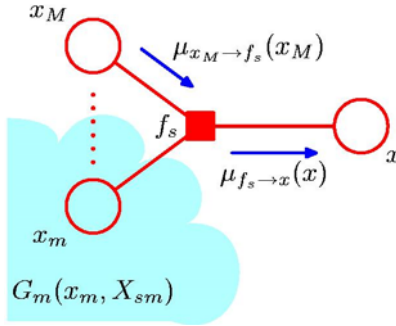
The Sum-Product Algorithm (4)



$$F_s(x, X_s) = f_s(x, \underline{x_1, \dots, x_M}) G_1(x_1, X_{s1}) \dots G_M(x_M, X_{sM})$$

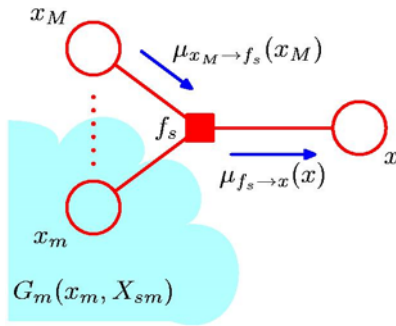
Other vars in the factor

The Sum-Product Algorithm (5)



$$\begin{aligned}\mu_{f_s \rightarrow x}(x) &= \sum_{x_1} \dots \sum_{x_M} f_s(x, x_1, \dots, x_M) \prod_{m \in \text{ne}(f_s) \setminus x} \left[\sum_{X_{sm}} G_m(x_m, X_{sm}) \right] \\ &= \sum_{x_1} \dots \sum_{x_M} f_s(x, x_1, \dots, x_M) \prod_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f_s}(x_m)\end{aligned}$$

The Sum-Product Algorithm (6)



$$\begin{aligned}\mu_{x_m \rightarrow f_s}(x_m) &\equiv \sum_{X_{sm}} G_m(x_m, X_{sm}) = \sum_{X_{sm}} \prod_{l \in \text{ne}(x_m) \setminus f_s} F_l(x_m, X_{ml}) \\ &= \prod_{l \in \text{ne}(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)\end{aligned}$$

The Sum-Product Algorithm (7)

- Initialization

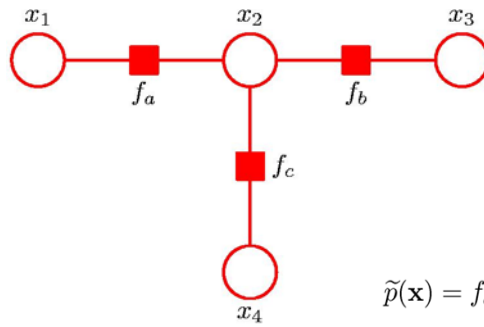


The Sum-Product Algorithm (8)

To compute local marginals:

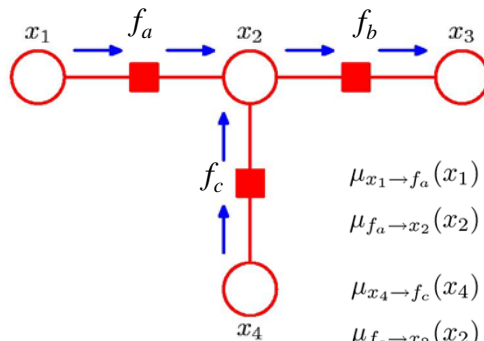
- Pick an arbitrary node as root
- Compute and propagate messages from the leaf nodes to the root, storing received messages at every node.
- Compute and propagate messages from the root to the leaf nodes, storing received messages at every node.
- Compute the product of received messages at each node for which the marginal is required, and normalize if necessary.

Sum-Product: Example (1)



$$\tilde{p}(\mathbf{x}) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4)$$

Sum-Product: Example (2)



$$\mu_{x_1 \rightarrow f_a}(x_1) = 1$$

$$\mu_{f_a \rightarrow x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2)$$

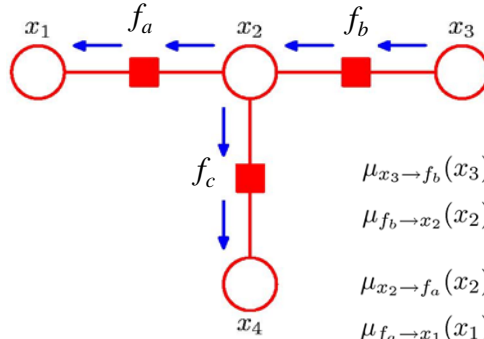
$$\mu_{x_4 \rightarrow f_c}(x_4) = 1$$

$$\mu_{f_c \rightarrow x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4)$$

$$\mu_{x_2 \rightarrow f_b}(x_2) = \mu_{f_a \rightarrow x_2}(x_2) \mu_{f_c \rightarrow x_2}(x_2)$$

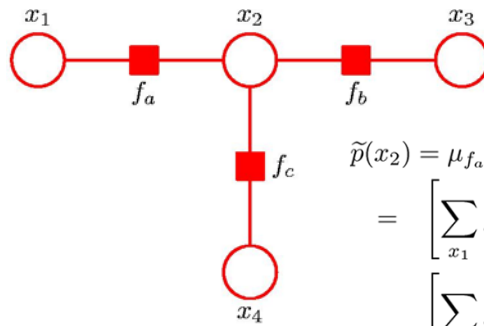
$$\mu_{f_b \rightarrow x_3}(x_3) = \sum_{x_2} f_b(x_2, x_3) \mu_{x_2 \rightarrow f_b}(x_2)$$

Sum-Product: Example (3)



$$\begin{aligned}
 \mu_{x_3 \rightarrow f_b}(x_3) &= 1 \\
 \mu_{f_b \rightarrow x_2}(x_2) &= \sum_{x_3} f_b(x_2, x_3) \\
 \mu_{x_2 \rightarrow f_a}(x_2) &= \mu_{f_b \rightarrow x_2}(x_2) \mu_{f_c \rightarrow x_2}(x_2) \\
 \mu_{f_a \rightarrow x_1}(x_1) &= \sum_{x_2} f_a(x_1, x_2) \mu_{x_2 \rightarrow f_a}(x_2) \\
 \mu_{x_2 \rightarrow f_c}(x_2) &= \mu_{f_a \rightarrow x_2}(x_2) \mu_{f_b \rightarrow x_2}(x_2) \\
 \mu_{f_c \rightarrow x_4}(x_4) &= \sum_{x_2} f_c(x_2, x_4) \mu_{x_2 \rightarrow f_c}(x_2)
 \end{aligned}$$

Sum-Product: Example (4)



$$\begin{aligned}
 \tilde{p}(x_2) &= \mu_{f_a \rightarrow x_2}(x_2) \mu_{f_b \rightarrow x_2}(x_2) \mu_{f_c \rightarrow x_2}(x_2) \\
 &= \left[\sum_{x_1} f_a(x_1, x_2) \right] \left[\sum_{x_3} f_b(x_2, x_3) \right] \\
 &\quad \left[\sum_{x_4} f_c(x_2, x_4) \right] \\
 &= \sum_{x_1} \sum_{x_3} \sum_{x_4} f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4) \\
 &= \sum_{x_1} \sum_{x_3} \sum_{x_4} \tilde{p}(\mathbf{x})
 \end{aligned}$$

Markov Random Fields: learning

CS 3750 Advanced Machine Learning

Markov random fields

- **Probabilistic models with symmetric dependences.**

- Typically models spatially varying quantities

$$P(x) \propto \prod_{c \in cl(x)} \phi_c(x_c)$$

$\phi_c(x_c)$ - A potential function (defined over factors)

- If $\phi_c(x_c)$ is strictly positive we can rewrite the definition as:

$$P(x) = \frac{1}{Z} \exp \left(- \sum_{c \in cl(x)} E_c(x_c) \right) \quad \text{- Energy function}$$

- Gibbs (Boltzman) distribution

$$Z = \sum_{x \in \{x\}} \exp \left(- \sum_{c \in cl(x)} E_c(x_c) \right) \quad \text{- A partition function}$$

CS 3750 Advanced Machine Learning

Types of Markov random fields

- **MRFs with discrete random variables**

- Clique potentials can be defined by mapping all clique-variable instances to \mathbb{R}
- Example: Assume two binary variables A,B with values $\{a1,a2,a3\}$ and $\{b1,b2\}$ are in the same clique c. Then:

$$\phi_c(A, B) \cong$$

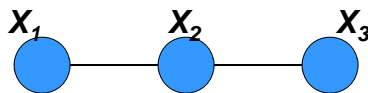
a1	b1	0.5
a1	b2	0.2
a2	b1	0.1
a2	b2	0.3
a3	b1	0.2
a3	b2	0.4

- Next: **Learning MRFs with discrete random vars**

CS 3750 Advanced Machine Learning

An example of MRF

- Undirected Graph



- Full joint distribution

$$p(X) = \frac{1}{Z} \psi_1(X_1, X_2) \cdot \psi_2(X_2, X_3) \cdot$$

- Parameters

$$\begin{aligned} &\psi_1(X_1 = 0, X_2 = 0), \psi_1(X_1 = 0, X_2 = 1), \\ &\psi_1(X_1 = 1, X_2 = 0), \psi_1(X_1 = 1, X_2 = 1), \\ &\psi_2(X_2 = 0, X_3 = 0), \psi_2(X_2 = 0, X_3 = 1), \\ &\psi_2(X_2 = 1, X_3 = 0), \psi_2(X_2 = 1, X_3 = 1). \end{aligned}$$

Assumptions

- Complete data set
 - No hidden variables, no missing value
 - Independent identically distribution (IID)
 - Discrete model
 - Known structure
 - Parameter independency
 - Maximum likelihood estimation
 - More difficult than that of Bayesian network
 - Decomposable or non-decomposable model
-

Notations

- V : set of nodes of the graph.
 - X_u : the random variable associated with $u \in V$
 x_u : an instantiation of X_u
 - C : a subset of V ,
 X_C : set of variables indexed by C
 x_c : an instantiation of X_C
 x_V or x : an instantiation of all random variables
 - N : number of samples in the data set D
 n : Index of data. $n = 1, 2, \dots, N$
 - $D : (D_1, D_2, \dots, D_N) = (x_{v,1}, x_{v,2}, \dots, x_{v,N})$
-

Maximum likelihood estimation for MRF

- Full joint distribution

$$p(x_V | \theta) = \frac{1}{Z} \prod_C \psi_C(x_C), \quad Z = \sum_{x_C} \prod_C \psi_C(x_C)$$

- Likelihood

$$p(D_n | \theta) = p(x_{V,n} | \theta) = \prod_{x_V} p(x_V | \theta)^{\delta(x_V, x_{V,n})}$$

$$\delta(x_V, x_{V,n}) = 1 \text{ iff } x_V = x_{V,n}$$

$$p(D | \theta) = \prod_n p(x_{V,n} | \theta) = \prod_n \prod_{x_V} p(x_V | \theta)^{\delta(x_V, x_{V,n})}$$

Maximum likelihood estimation for MRF

- Log likelihood

$$\begin{aligned} l(\theta, D) &= \log p(D | \theta) = \log \left(\prod_n \prod_{x_V} p(x_V | \theta)^{\delta(x_V, x_{V,n})} \right) \\ &= \sum_n \sum_{x_V} \delta(x_V, x_{V,n}) \log p(x_V | \theta) = \sum_{x_V} m(x_V) \log p(x_V | \theta) \end{aligned}$$

- Count: the number of times that configuration x_V is observed is defined as:

$$m(x_V) \equiv \sum_n \delta(x_V, x_{V,n})$$

- And marginal count for clique C :

$$m(x_C) \equiv \sum_{x_V \setminus C} m(x_V)$$

Count and Marginal Count

X_1	X_2	X_3
0	0	0
0	0	1
1	1	0
1	0	1
0	0	1
1	0	1
1	1	1
0	0	1
1	0	0
0	1	0

$$m((X_1=0, X_2=0, X_3=1)) = ?$$

$$m((X_1=1, X_2=0)) = ?$$

Count and Marginal Count

X_1	X_2	X_3
0	0	0
0	0	1
1	1	0
1	0	1
0	0	1
1	0	1
1	1	1
0	0	1
1	0	0
0	1	0

$$m((X_1=0, X_2=0, X_3=1)) = 3$$

$$m((X_1=1, X_2=0)) = ?$$

Count and Marginal Count

X_1	X_2	X_3
0	0	0
0	0	1
1	1	0
1	0	1
0	0	1
1	0	1
1	1	1
0	0	1
1	0	0
0	1	0

$$m((X_1=0, X_2=0, X_3=1)) = 3$$

$$m((X_1=1, X_2=0)) = 3$$

Maximum likelihood estimation for MRF

- Log likelihood

$$l(\theta, D)$$

$$= \sum_n \sum_{x_V} \delta(x_V, x_{V,n}) \log p(x_V | \theta)$$

$$= \sum_{x_V} m(x_V) \log p(x_V | \theta)$$

$$= \sum_{x_V} m(x_V) \log \left(\frac{1}{Z} \prod_C \psi_C(x_C) \right)$$

$$= \sum_{x_V} m(x_V) \sum_C \log \psi_C(x_C) - \sum_{x_V} m(x_V) \log Z$$

$$= \sum_C \sum_{x_C} m(x_C) \log \psi_C(x_C) - N \log Z$$

Bayesian network vs MRF

- Bayesian network

Parameters are decomposed

$$l(\theta, D) = \sum_u \sum_{x_{\{u\} \cup pa(u)}} m(x_{\{u\} \cup pa(u)}) \log \theta_u(x_{\{u\} \cup pa(u)})$$

- MRF

Parameters are **not** decomposed

$$l(\theta, D) = \sum_C \sum_{x_C} m(x_C) \log \psi_C(x_C) - N \log Z$$

$$\log Z = \log \sum_{x_C} \prod_C \psi_C(x_C)$$
