11. (Probabilistic) Graphical Models

Dongwoo Kim

dongwookim@postech.ac.kr

CSED515 - 2023 Spring

Statistical Graphical Model

A statistical model is a set of assumptions to explain/understand the generation of sample data.

We often assume some dependences among random variables

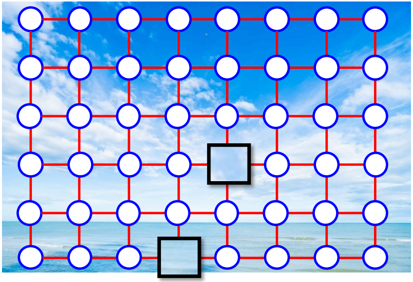
Graphical models visualize such dependences efficiently and provide a set of efficient machine learning tools, e.g., sum-product and max-product belief propagation for ML/MAP.

Dependence (1): Correlation

An efficient inference may use not only value of variable but also relation among variables

- e.g., binary classification of image tiles: sky or sea?
- We need compositive inference based on correlation rather than individual ones without correlation
- The correlation can be represented by an undirected graph

Sky or Sea?



Dependence (2): Casuality

To estimate a latent variable from observed variables, we often construct and use causality model to connect them

- e.g., reasoning slipped on the step
- **R**: rain, **Sp**: sprinkler, **W**: wet, **Sl**: slipped
- The dependence can be represented by directed graph

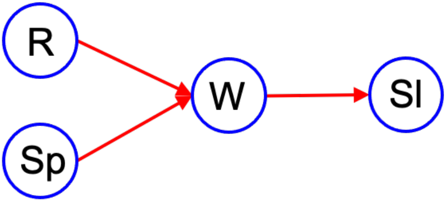


Table of Contents

Table of Contents

Directed Graphical Model a.k.a. Bayesian Network

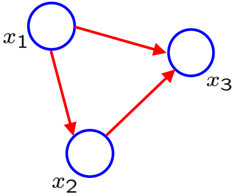
Directed acyclic graph (DAG) to describe the joint probability of all random variables, where

- Nodes represent random variables, and edges represent causal relationships, i.e.,

$$p(x_1, \ldots, x_N) = \prod_{i \in [N]} p(x_i \mid \mathrm{pa}(x_i))$$

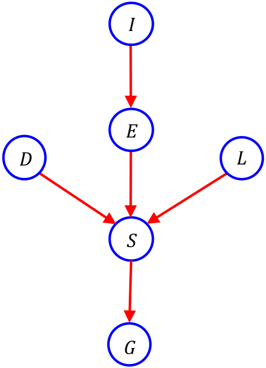
  where $\mathrm{pa}(x_i)$ denotes the set of x_i's every parent.

- No cycle is allowed.



$$p(x_1, x_2, x_3) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2)$$

Factorization in Directed Graphs



The joint probability is factorized as follows:

$$p(L, D, I, E, S, G) \\ = p(L)p(D)p(I)p(E \mid I)p(S \mid D, L, E)p(G \mid S)$$

where we consider binary random variables: **L**ecture quality, **D**ifficulty, **I**ntelligence, **E**fforts, **S**core, and **G**rade.

- Originally, a table of size $2^6 - 1 = 63$ is required at least.
- By factorization, the table size can be reduced to $1+1+1+2+8+2 = 15$.

Fewer edges not only reduce the parameter number more but also provide more information.

Undirected Graph a.k.a. Markov Random Feild

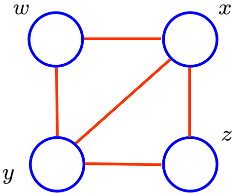
The joint distribution is the product of non-negative functions over the maximal cliques of the undirected graph

$$p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

where $x = \{x_i\}_{i \in [N]}$, $\mathcal{C} \subset 2^{\{N\}}$ is the set of all maximal cliques, the clique potential $\psi_C(x_C)$ is a non-negative function, which represents correlation among $x_C = \{x_i\}_{i \in C}$, and Z is the normalization constant:

$$Z = \sum_x \prod_{C \in \mathcal{C}} \psi_C(x_C) \, .$$

Example of Cliques



$$p(w,x,y,z) = \frac{1}{Z}\psi_{wxy}(w,x,y)\psi_{xyz}(x,y,z)$$

- Clique: a fully connected subset of a graph, e.g.,

$$wx, wy, yz, xz, xyz, wxy, xy \ .$$

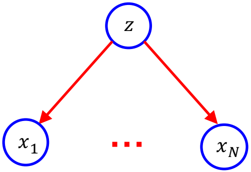
- Maximal clique: a clique that is not a part of another cliques, e.g.,

$$xyz, wxy \ .$$

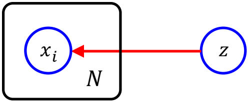
Convenient Notation (1)

- Plate for sequence of variables

$$p(\{x_i\}_{i\in[N]}, z) = p(z) \prod_{i\in[N]} p(x_i \mid z)$$



or



Convenient Notation (2)

- Filled nodes for visible or observed variables, while empty ones for latent variables, e.g.,

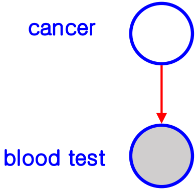


Table of Contents

Conditional Independence

- X and Y are conditional independent given Z, and denote

$$X \perp\!\!\!\perp Y \mid Z$$

if and only if for all possible values (x, y, z) of (X, Y, Z),

$$\begin{aligned} p(x, y \mid z) &= p(x \mid z) p(y \mid z) \\ &\iff p(x \mid y, z) = p(x \mid z) \quad \text{or} \quad p(y \mid z) = 0 \,. \end{aligned}$$

- X and Y are (marginal) independent and denote

$$X \perp\!\!\!\perp Y \quad (\iff X \perp\!\!\!\perp Y \mid \emptyset)$$

if and only if for all possible values (x, y) of (X, Y),

$$p(x, y) = p(x) p(y) \quad .$$

Examples of Conditional Independence

- (amount of speeding fine) $\perp\!\!\!\perp$ (type of car) | (speed)
- (lung cancer) $\perp\!\!\!\perp$ (yellow teeth) | (smoking)
- Even if abilities of teams A and B are (marginal) independent, i.e.,

  $$\text{(ability of team A)} \perp\!\!\!\perp \text{(ability of team B)} \mid \emptyset$$

  or simply, (ability of team A) $\perp\!\!\!\perp$ (ability of team B),
  a conditional independence given some event may not hold,

  $$\text{(ability of team A)} \not\perp\!\!\!\perp \text{(ability of team B)} \mid \text{(winner of A vs B)} .$$

Conditional Independence in Graphical Model

The conditional independence gives some hints on what we should observe and how to learn, e.g.,

$$\text{(lung cancer)} \perp\!\!\!\perp \text{(yellow teeth)} \mid \text{(smoking)}\ .$$

Graphical model is useful to describe the conditional independence

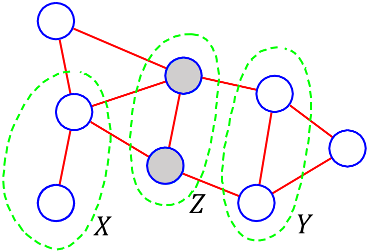
- Undirected graph: straightforward
- Directed graph: somewhat subtle

Markov Property in Undirected Graph

Conditional independence given by graph separation:

- Consider all paths from X to Y and if all such paths through one or more nodes in Z then paths are blocked and we have the following conditional independence:

$$X \perp\!\!\!\perp Y \mid Z \, .$$

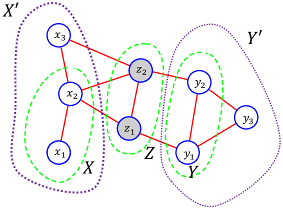


Hammersley-Clifford Theorem (1971)

Theorem

A probability function p formed by a normalized product of positive functions on cliques of undirected graph $G = (V, E)$ is a Markov field relative to G, i.e., for any vertex subset $X, Y, Z \subset V$, if Z separates between X and Y, then $X \perp\!\!\!\perp Y \mid Z$.

Proof of Hammersley-Clifford Theorem (1)



$$
\begin{aligned}
p(X', Y' \mid Z) &= p(X' \mid Z) p(Y' \mid Z) \\
\implies p(X, Y \mid Z) &= \sum_{x_3} \sum_{y_3} p(X, x_3 \mid Z) p(Y, y_3 \mid Z) \\
&= \left(\sum_{x_3} p(X, x_3 \mid Z) \right) \left(\sum_{y_3} p(Y, y_3 \mid Z) \right) \\
&= p(X \mid Z) p(Y \mid Z)
\end{aligned}
$$

Let X' and Y' be the disjoint components separated by Z such that $X \subset X'$ and $Y \subset Y'$. It suffices to show the conditional independence of X' and Y' given Z.

Proof of Hammersley-Clifford Theorem (2)

Let X' and Y' be the disjoint components separated by Z such that $X \subset X'$ and $Y \subset Y'$. It suffices to show the conditional independence of X' and Y' given Z.

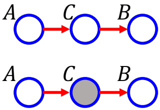
Noting that the set $\mathcal{C}$ of cliques can be partitioned into three disjoint sets: $\mathcal{C}_1 = \{C \in \mathcal{C} : X' \cap C \neq \emptyset\}$, $\mathcal{C}_2 = \{C \in \mathcal{C} : Y' \cap C \neq \emptyset\}$, and $\mathcal{C}_3 = \{C \in \mathcal{C} : Z \cap C \subset Z\}$,

$$
\begin{aligned}
p(X', Y', Z) &\propto \prod_{C \in \mathcal{C}} \psi_C(x_C, y_C, z_C) \\
&= \prod_{i=1,2,3} \left(\prod_{C \in \mathcal{C}_i} \psi_C(x_C, y_C, z_C) \right) ,
\end{aligned}
$$

which completes the proof since the product terms for $i = 1, 2$ are functions of either (X', Z) or (Y', Z), respectively.

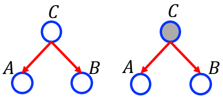
Markov Property in Directed Graph

- Head-to-tail: $p(A, B, C) = p(A)p(C \mid A)p(B \mid C)$



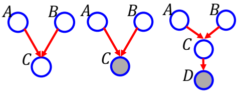
$$\begin{aligned} p(A, B \mid C) &= \frac{p(A)p(C \mid A)p(B \mid C)}{p(C)} \\ &= p(A \mid C)p(B \mid C) \end{aligned} \qquad \Longrightarrow \quad A \perp\!\!\!\perp B \mid C$$

- Tail-to-tail: $p(A, B, C) = p(C)p(A \mid C)p(B \mid C)$



$$\begin{aligned} p(A, B \mid C) &= \frac{p(C)p(A \mid C)p(B \mid C)}{p(C)} \\ &= p(A \mid C)p(B \mid C) \end{aligned} \qquad \Longrightarrow \quad A \perp\!\!\!\perp B \mid C$$

- Head-to-head: $p(A, B, C) = p(A)p(B)p(C \mid A, B)$



$$\begin{aligned} p(A, B) &= p(A)p(B) \sum_{C} p(C \mid A, B) \\ &= p(A)p(B) \end{aligned} \qquad \Longrightarrow \quad A \perp\!\!\!\perp B$$

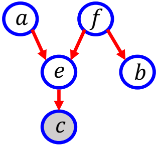
Note: an observation of any co-descendant of A and B creates dependence, i.e., $A \not\!\perp\!\!\!\perp B \mid C$

D-Separation

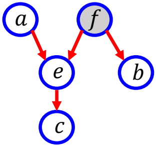
To determine whether a particular conditional independence statement ($A \perp\!\!\!\perp B \mid C$) in a given DAG:

- Consider all possible paths from any node in A to any node in B and determine whether the path is blocked by C:
  - The arrows on the path meet either head-to-tail or tail-to-tail at a node in C
  - The arrows on the path meet head-to-head at a node which is neither a member of C nor any of its descendants is in C
- Conditional independence iff all possible paths are blocked

Examples:



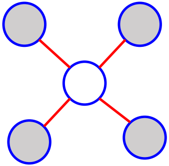
$a \not\perp\!\!\!\perp b | c$



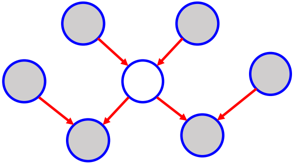
$a \perp\!\!\!\perp b | f$

Markov Blankets

- $C \subset V$ is a Markov blanket for $a \in V$ iff $a \perp\!\!\!\perp b \mid C$ for any $b \notin \{a\} \cup C$.
- A minimal Markov blanket is a Markov boundary, e.g.,



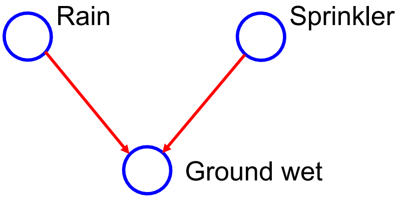
Neighboring nodes



Parents, children, co-parents

Explaining Away

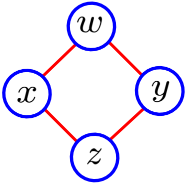
One may dislike using directed graphs with the confusing head-to-head relationship, but, in fact, it can be a main reason for using it:



- Rain and sprinkler are independent (given nothing), but conditionally dependent given the ground wet.

Undirected vs. Directed

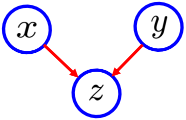
No directed acyclic graph can represent these and only these independencies:



$$x \perp\!\!\!\perp y \mid w, z$$

$$w \perp\!\!\!\perp z \mid x, y$$

No undirected graph can represent these and only these independence and dependence:

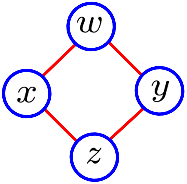


$$x \perp\!\!\!\perp y \mid \emptyset$$

$$x \not\!\perp\!\!\!\perp y \mid z$$

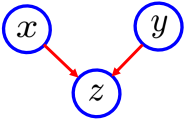
Graphical Models

Undirected only



$$x \perp\!\!\!\perp y \mid w, z$$
$$w \perp\!\!\!\perp z \mid x, y$$

Directed only



$$x \perp\!\!\!\perp y \mid \emptyset$$
$$x \not\!\perp\!\!\!\perp y \mid z$$

Graphical model is nothing but visualization of factorization of joint probability

- Directed: $p(x_1, \ldots, x_N) = \prod_{i \in [N]} p(x_i \mid \mathrm{pa}(x_i))$
- Undirected: $p(x_1, \ldots, x_N) = \prod_{C \in \mathcal{C}} \psi_C(x_C)$

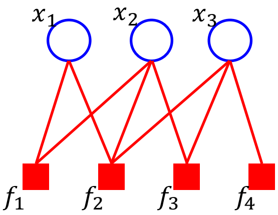
Table of Contents

Factor Graph

A (undirected) bipartite graph between variable nodes $\{x_i\}_{i\in[N]}$ and factor nodes $\{f_j\}_{j\in[M]}$:

$$p(x_1, \ldots, x_N) = \prod_{j\in[M]} f_j(\mathrm{nei}_j)$$

where each factor f_j is a function of variable nodes nei_j connected to f_j.

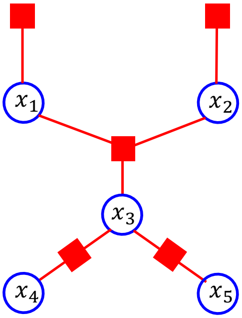


$$p(x_1, x_2, x_3) = f_1(x_1, x_2)f_2(x_1, x_2, x_3)f_3(x_2, x_3)f_4(x_3)$$

Conditional Independence in Factor Graph

To check $x \perp\!\!\!\perp y \mid z$,

- Consider all paths from x to y
- Check if every path is blocked by z

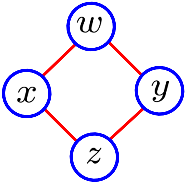


$$x_1 \perp\!\!\!\perp x_5 \mid x_3 \,, \quad x_1 \not\perp\!\!\!\perp x_2 \mid x_3$$

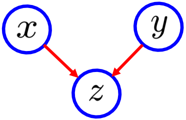
Factor Graph Specialized for Factorization (1)

Factor graph generalizes undirected and directed graphical model in terms of expression power

- A factor in undirected graph has to be assigned to a maximal clique
- A directed graphical model does not have cycles



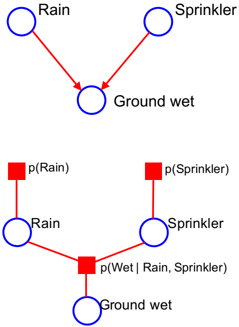
$$x \perp\!\!\!\perp y \mid w, z$$
$$w \perp\!\!\!\perp z \mid x, y$$



$$x \perp\!\!\!\perp y \mid \emptyset$$
$$x \not\perp\!\!\!\perp y \mid z$$

Factor Graph Specialized for Factorization (2)

Note that directed and undirected graphical models may provide more insights on relationships among variables.



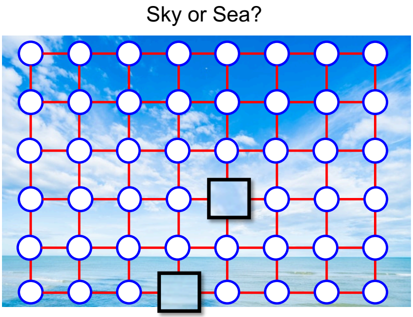


Table of Contents

Two Important Problems

- Marginalization

$$p(x_i \mid \mathcal{D}) = \sum_{x_{-i}} p(x_1, \ldots, x_N \mid \mathcal{D})$$

where $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N)$.

- Maximization

$$(\hat{x}_1, \ldots, \hat{x}_N) = \arg\max_{x_1, \ldots, x_N} p(x_1, \ldots, x_N \mid \mathcal{D}) \; .$$

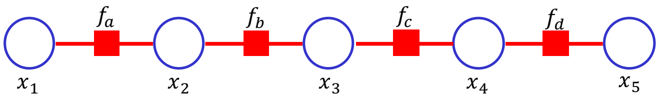
A naive method for marginalization or maximization requires exponentially many ($O(L^{N-1})$) summations or comparisons in N.

An Example of Factorization

Consider a joint probability of discrete random variables factorized as:

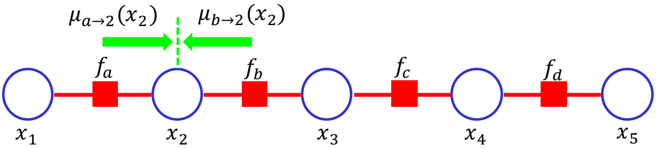
$$p(x_1, \ldots, x_5) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_3, x_4) f_d(x_4, x_5) \ ,$$

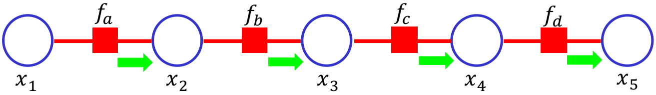
of which factor graph is:



The marginal probability of x_2 can be calculated...

$$
\begin{aligned}
p(x_2) &= \sum_{x_1}\sum_{x_3}\sum_{x_4}\sum_{x_5} f_a(x_1,x_2) f_b(x_2,x_3) f_c(x_3,x_4) f_d(x_4,x_5) \\
&= \left(\sum_{x_1} f_a(x_1,x_2)\right)\left(\sum_{x_3}\sum_{x_4}\sum_{x_5} f_b(x_2,x_3) f_c(x_3,x_4) f_d(x_4,x_5)\right) \\
&= \left(\sum_{x_1} f_a(x_1,x_2)\right)\left(\sum_{x_3}\sum_{x_4} f_b(x_2,x_3) f_c(x_3,x_4) \sum_{x_5} f_d(x_4,x_5)\right) \\
&= \underbrace{\left(\sum_{x_1} f_a(x_1,x_2)\right)}_{\stackrel{\text{def}}{=} \mu_{a\to 2}(x_2)}\underbrace{\left(\sum_{x_3} f_b(x_2,x_3) \sum_{x_4} f_c(x_3,x_4) \sum_{x_5} f_d(x_4,x_5)\right)}_{\stackrel{\text{def}}{=} \mu_{b\to 2}(x_2)}
\end{aligned}
$$





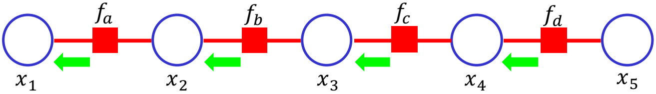
$$p(x_1) = \underbrace{\left(\sum_{x_2} f_a(x_1, x_2) \sum_{x_3} f_b(x_2, x_3) \sum_{x_4} f_c(x_3, x_4) \sum_{x_5} f_d(x_4, x_5) \right)}_{\stackrel{\text{def}}{=} \mu_{a \to 1}(x_1)}$$

$$p(x_2) = \underbrace{\left(\sum_{x_1} f_a(x_1, x_2) \right)}_{\stackrel{\text{def}}{=} \mu_{a \to 2}(x_2)} \underbrace{\left(\sum_{x_3} f_b(x_2, x_3) \sum_{x_4} f_c(x_3, x_4) \sum_{x_5} f_d(x_4, x_5) \right)}_{\stackrel{\text{def}}{=} \mu_{b \to 2}(x_2)}$$

$$p(x_3) = \underbrace{\left(\sum_{x_2} f_b(x_2, x_3) \mu_{a \to 2}(x_2) \right)}_{\stackrel{\text{def}}{=} \mu_{b \to 3}(x_3)} \underbrace{\left(\sum_{x_4} f_c(x_3, x_4) \sum_{x_5} f_d(x_4, x_5) \right)}_{\stackrel{\text{def}}{=} \mu_{c \to 3}(x_3)}$$

$$p(x_4) = \underbrace{\left(\sum_{x_3} f_c(x_3, x_4) \mu_{b \to 3}(x_3) \right)}_{\stackrel{\text{def}}{=} \mu_{c \to 4}(x_4)} \underbrace{\left(\sum_{x_5} f_d(x_4, x_5) \right)}_{\stackrel{\text{def}}{=} \mu_{d \to 4}(x_4)}$$

$$p(x_5) = \underbrace{\left(\sum_{x_4} f_d(x_4, x_5) \mu_{c \to 4}(x_4) \right)}_{\stackrel{\text{def}}{=} \mu_{d \to 5}(x_4)}$$



$$p(x_1) = \underbrace{\left(\sum_{x_2} f_a(x_1, x_2) \sum_{x_3} f_b(x_2, x_3) \sum_{x_4} f_c(x_3, x_4) \sum_{x_5} f_d(x_4, x_5) \right)}_{\stackrel{\text{def}}{=} \mu_{a\to 1}(x_1) = \sum_{x_2} f_a(x_1, x_2)\mu_{b\to 2}(x_2)}$$

$$p(x_2) = \underbrace{\left(\sum_{x_1} f_a(x_1, x_2) \right)}_{\stackrel{\text{def}}{=} \mu_{a\to 2}(x_2)} \underbrace{\left(\sum_{x_3} f_b(x_2, x_3) \sum_{x_4} f_c(x_3, x_4) \sum_{x_5} f_d(x_4, x_5) \right)}_{\stackrel{\text{def}}{=} \mu_{b\to 2}(x_2) = \sum_{x_3} f_b(x_2, x_3)\mu_{c\to 3}(x_3)}$$

$$p(x_3) = \underbrace{\left(\sum_{x_2} f_b(x_2, x_3)\mu_{a\to 2}(x_2) \right)}_{\stackrel{\text{def}}{=} \mu_{b\to 3}(x_3)} \underbrace{\left(\sum_{x_4} f_c(x_3, x_4) \sum_{x_5} f_d(x_4, x_5) \right)}_{\stackrel{\text{def}}{=} \mu_{c\to 3}(x_3) = \sum_{x_4} f_c(x_3, x_4)\mu_{d\to 4}(x_4)}$$

$$p(x_4) = \underbrace{\left(\sum_{x_3} f_c(x_3, x_4)\mu_{b\to 3}(x_3) \right)}_{\stackrel{\text{def}}{=} \mu_{c\to 4}(x_4)} \underbrace{\left(\sum_{x_5} f_d(x_4, x_5) \right)}_{\stackrel{\text{def}}{=} \mu_{d\to 4}(x_4)}$$

$$p(x_5) = \underbrace{\left(\sum_{x_4} f_d(x_4, x_5)\mu_{c\to 4}(x_4) \right)}_{\stackrel{\text{def}}{=} \mu_{d\to 5}(x_4)}$$

An Efficient Marginalization via Factorization

Consider a joint probability of $\{x_i \in [L]\}_{i \in [N]}$ factorized as:

$$p(x_1, \dots, x_N) = f_1(x_1, x_2) f_2(x_2, x_3) \dots f_{N-1}(x_{N-1}, x_N) \; .$$

An efficient computation of marginal probability of x_i is:

$$p(x_i) = \mu_i^-(x_i) \mu_i^+(x_i) \; ,$$

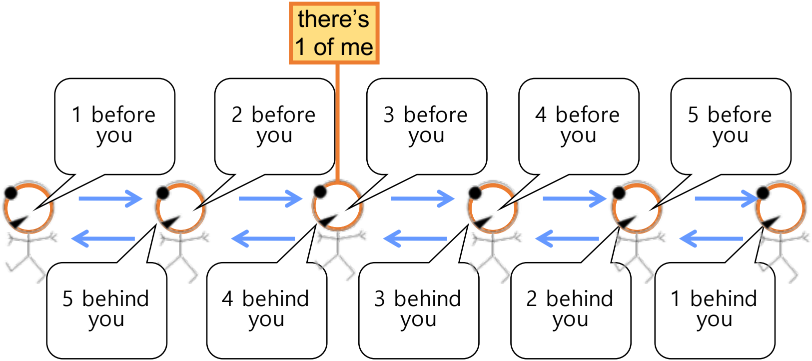
where $\mu_1^-(x_1) \stackrel{\text{def}}{=} 1$, $\mu_N^+(x_N) \stackrel{\text{def}}{=} 1$,

$$\mu_i^-(x_i) \stackrel{\text{def}}{=} \sum_{x_{i-1}} f_{i-1}(x_{i-1}, x_i) \mu_{i-1}^-(x_{i-1}) = \sum_{x_{i-1}} f_{i-1}(x_{i-1}, x_i) \sum_{x_{i-2}} f_{i-2}(x_{i-2}, x_{i-1}) \dots \sum_{x_1} f_1(x_1, x_2)$$

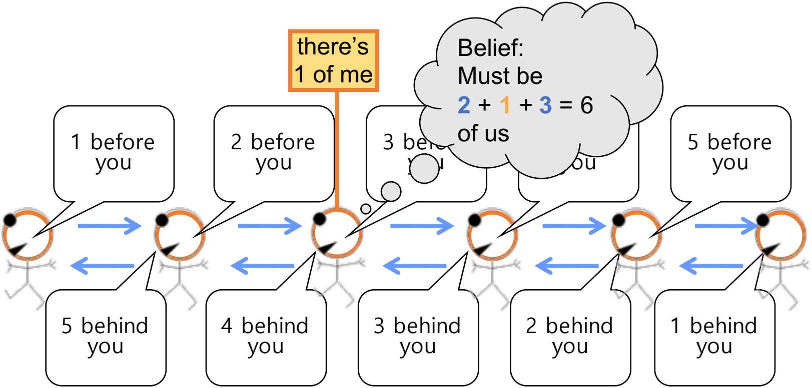
$$\mu_i^+(x_i) \stackrel{\text{def}}{=} \sum_{x_{i+1}} f_i(x_i, x_{i+1}) \mu_{i+1}^+(x_{i+1}) = \sum_{x_{i+1}} f_i(x_i, x_{i+1}) \sum_{x_{i+2}} f_{i+2}(x_{i+2}, x_{i+3}) \dots \sum_{x_N} f_{N-1}(x_{N-1}, x_N) \; .$$

The number of summation is reduced from $O(L^{N-1})$ (exponential in N) to $O((N-1) \cdot L^2)$ (polynomial in N).

An Intuitive Understanding



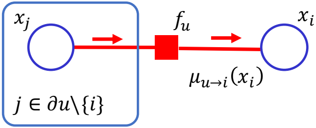
An Intuitive Understanding



Sum-Product Belief Propagation (BP) in Tree

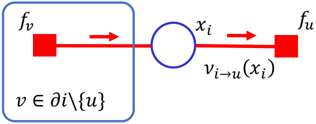
Denoting $\partial(\text{node}) \stackrel{\text{def}}{=}$ (the set of neighbors), and $x_I \stackrel{\text{def}}{=} \{x_i\}_{i \in I}$,

- From factor f_u to variable x_i



$$\mu_{u \to i}(x_i) \stackrel{\text{def}}{=} \sum_{x_{\partial u \backslash \{i\}}} f_u(x_{\partial u}) \prod_{j \in \partial u \backslash \{i\}} \nu_{j \to u}(x_j)$$

- From variable x_i to factor f_u



$$\nu_{i \to u}(x_i) \stackrel{\text{def}}{=} \prod_{v \in \partial i \backslash \{u\}} \mu_{v \to i}(x_i)$$

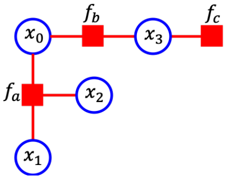
- Marginal probability $p(x_i) = \prod_{v \in \partial i} \mu_{v \to i}(x_i)$.

Scheduling in Sum-Product BP on Tree

$$\mu_{u\to i}(x_i) \stackrel{\text{def}}{=} \sum_{x_{\partial u \setminus \{i\}}} f_u(x_{\partial u}) \prod_{j \in \partial u \setminus \{i\}} \nu_{j\to u}(x_j)$$

$$\nu_{i\to u}(x_i) \stackrel{\text{def}}{=} \prod_{v \in \partial i \setminus \{u\}} \mu_{v\to i}(x_i)$$

$$p(x_i) = \prod_{v \in \partial i} \mu_{v\to i}(x_i)$$



where the product over empty set is 1, i.e., we will start from leaves:

- Leaf variable x_i, e.g., x_1, x_2, with $\partial i = \{u\}$:

$$\nu_{i\to u}(x_i) = \prod_{v \in \partial i \setminus \{u\}} \mu_{v\to i}(x_i) = 1 \; .$$

- Leaf factor f_u, e.g., f_c, with $\partial u = \{i\}$:

$$\mu_{u\to i}(x_i) = \sum_{x_{\partial u \setminus \{i\}}} f_u(x_{\partial u}) \prod_{j \in \partial u \setminus \{i\}} \nu_{j\to u}(x_j) = f_u(x_i)$$

Example of Sum-Product BP (1)

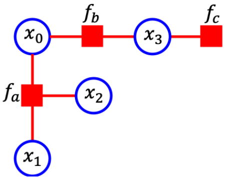
Starting from leaves:

Factor to variable

$$\mu_{u\to i}(x_i) \stackrel{\text{def}}{=} \sum_{x_{\partial u}\setminus\{i\}} f_u(x_{\partial u}) \prod_{j\in\partial u\setminus\{i\}} \nu_{j\to u}(x_j)$$

Variable to factor

$$\nu_{i\to u}(x_i) \stackrel{\text{def}}{=} \prod_{v\in\partial i\setminus\{u\}} \mu_{v\to i}(x_i)$$



$$\begin{aligned}
\mu_{c\to 3}(x_3) &= f_c(x_3)\\
\nu_{1\to a}(x_1) &= 1\\
\nu_{2\to a}(x_2) &= 1\\
\nu_{3\to b}(x_3) &= \mu_{c\to 3}(x_3) = f_c(x_3)\\
\mu_{a\to 0}(x_0) &= \sum_{x_1,x_2} f_a(x_0,x_1,x_2)\times 1\times 1\\
\mu_{b\to 0}(x_0) &= \sum_{x_3} f_b(x_0,x_3)f_c(x_3)
\end{aligned}$$

$$\begin{aligned}
\nu_{0\to b}(x_0) &= \sum_{x_1,x_2} f_a(x_0,x_1,x_2)\\
\nu_{0\to a}(x_0) &= \sum_{x_3} f_b(x_0,x_3)f_c(x_3)\\
\mu_{b\to 3}(x_3) &= \sum_{x_0} f_b(x_0,x_3)\left(\sum_{x_1,x_2} f_a(x_0,x_1,x_2)\right)\\
\mu_{a\to 1}(x_1) &= \sum_{x_0,x_2} f_a(x_0,x_1,x_2)\Big(1\times\sum_{x_3} f_b(x_0,x_3)f_c(x_3)\Big)\\
\mu_{a\to 2}(x_2) &= \sum_{x_0,x_1} f_a(x_0,x_1,x_2)\Big(1\times\sum_{x_3} f_b(x_0,x_3)f_c(x_3)\Big)
\end{aligned}$$

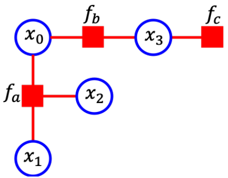
Example of Sum-Product BP (2)

Aggregation:

$$p(x_i) = \prod_{v \in \partial i} \mu_{v \to i}(x_i) \ ,$$

$$p(x_0) = \mu_{a \to 0}(x_0) \mu_{b \to 0}(x_0)$$

$$= \left(\sum_{x_1, x_2} f_a(x_0, x_1, x_2) \right) \left(\sum_{x_3} f_b(x_0, x_3) f_c(x_3) \right) \ .$$



$$\mu_{c \to 3}(x_3) = f_c(x_3)$$

$$\nu_{1 \to a}(x_1) = 1$$

$$\nu_{2 \to a}(x_2) = 1$$

$$\nu_{3 \to b}(x_3) = \mu_{c \to 3}(x_3) = f_c(x_3)$$

$$\mu_{a \to 0}(x_0) = \sum_{x_1, x_2} f_a(x_0, x_1, x_2) \times 1 \times 1$$

$$\mu_{b \to 0}(x_0) = \sum_{x_3} f_b(x_0, x_3) f_c(x_3)$$

$$\nu_{0 \to b}(x_0) = \sum_{x_1, x_2} f_a(x_0, x_1, x_2)$$

$$\nu_{0 \to a}(x_0) = \sum_{x_3} f_b(x_0, x_3) f_c(x_3)$$

$$\mu_{b \to 3}(x_3) = \sum_{x_0} f_b(x_0, x_3) \left(\sum_{x_1, x_2} f_a(x_0, x_1, x_2) \right)$$

$$\mu_{a \to 1}(x_1) = \sum_{x_0, x_2} f_a(x_0, x_1, x_2) \left(1 \times \sum_{x_3} f_b(x_0, x_3) f_c(x_3) \right)$$

$$\mu_{a \to 2}(x_2) = \sum_{x_0, x_1} f_a(x_0, x_1, x_2) \left(1 \times \sum_{x_3} f_b(x_0, x_3) f_c(x_3) \right)$$

Loopy Sum-Product BP

Sum-product BP is exact on tree, i.e., cycle free, but it is applicable even if there are loops, while we have no guarantee (exactness or convergence) in general.

Loopy Sum-Product BP algorithm

- Arbitrary initialization of messages $\mu^{(0)}$ and $\nu^{(0)}$
- Iterative update of messages $\mu^{(k)}$ and $\nu^{(k)}$ for $k = 1, 2, \ldots,$
  - Factor to variable

$$\mu_{u \to i}^{(k+1)}(x_i) \propto \sum_{x_{\partial u \setminus \{i\}}} f_u(x_{\partial u}) \prod_{j \in \partial u \setminus \{i\}} \nu_{j \to u}^{(k)}(x_i) , \qquad \forall u, i .$$

  - Variable to factor

$$\nu_{i \to u}(x_i) \propto \prod_{v \in \partial i \setminus \{u\}} \mu_{v \to i}^{(k)}(x_i) , \qquad \forall u, i .$$

- Computation of belief as an approximation of marginal probability

$$b_i^{(k)}(x_i) \propto \prod_{v \in \partial i} \mu_{v \to i}^{(k)}(x_i) , \qquad \forall i .$$

Table of Contents

Maximization via Factorization

Consider

$$(\hat{x}_1, \ldots, \hat{x}_5) = \arg\max_{x_1,\ldots,x_5} p(x_1, \ldots, x_5) ,$$

where the joint probability is factorized as:

$$p(x_1, \ldots, x_5) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_3, x_4) f_d(x_4, x_5) .$$

The maximization can be done efficiently as:

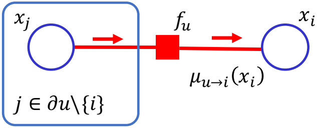
$$\begin{aligned}
&\max_{x_1,x_2,x_3,x_4,x_5} \{f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_3, x_4) f_d(x_4, x_5)\} \\
&= \max_{x_1,x_2,x_3,x_4} \left\{ f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_3, x_4) \max_{x_5} \{f_d(x_4, x_5)\} \right\} \\
&= \max_{x_2} \left\{ \max_{x_1} \{f_a(x_1, x_2)\} \max_{x_3} \left\{ f_b(x_2, x_3) \max_{x_4} \left\{ f_c(x_3, x_4) \max_{x_5} \{f_d(x_4, x_5)\} \right\} \right\} \right\} ,
\end{aligned}$$

which looks pretty similar to the marginalization via factorization.

Max-Product Belief Propagation in Tree

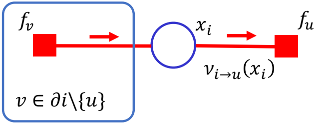
Denoting $\partial(\text{node}) \stackrel{\text{def}}{=}$ (the set of neighbors), and $x_I \stackrel{\text{def}}{=} \{x_i\}_{i \in I}$,

- From factor f_u to variable x_i



$$\mu_{u\to i}(x_i) \stackrel{\text{def}}{=} \max_{x_{\partial u \backslash \{i\}}} \left\{ f_u(x_{\partial u}) \prod_{j \in \partial u \backslash \{i\}} \nu_{j \to u}(x_j) \right\}$$

- From variable x_i to factor f_u



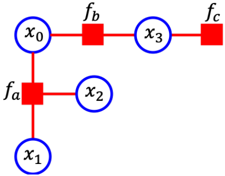
$$\nu_{i\to u}(x_i) \stackrel{\text{def}}{=} \prod_{v \in \partial i \backslash \{u\}} \mu_{v \to i}(x_i)$$

- Maximum probability $p_{\max} = \max_{x_i} \left\{ \prod_{v \in \partial i} \mu_{v \to i}(x_i) \right\}$, of which maximizer $\hat{x}_i$ forms the most likely configuration.

Scheduling in Max-Product BP on Tree

$$\mu_{u\to i}(x_i) \stackrel{\text{def}}{=} \max_{x_{\partial u \setminus \{i\}}} \left\{ f_u(x_{\partial u}) \prod_{j \in \partial u \setminus \{i\}} \nu_{j\to u}(x_i) \right\}$$

$$\nu_{i\to u}(x_i) \stackrel{\text{def}}{=} \prod_{v \in \partial i \setminus \{u\}} \mu_{v\to i}(x_i)$$



We start from leaves:

- Leaf variable x_i, e.g., x_1, x_2, with $\partial i = \{u\}$:

$$\mu_{i\to u}(x_i) = \prod_{v \in \partial i \setminus \{u\}} \nu_{v\to i}(x_i) = 1 \; .$$

- Leaf factor f_u, e.g., f_c, with $\partial u = \{i\}$:

$$\nu_{u\to i}(x_i) = \max_{x_{\partial u \setminus \{i\}}} \left\{ f_u(x_{\partial u}) \prod_{j \in \partial u \setminus \{i\}} \mu_{j\to u}(x_i) \right\} = f_u(x_i)$$

Example of Max-Product BP (1)

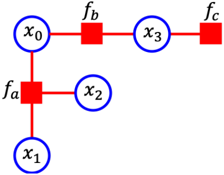
Starting from leaves:

Factor to variable

$$\mu_{u\to i}(x_i) \stackrel{\text{def}}{=} \max_{x_{\partial u \setminus \{i\}}} \left\{ f_u(x_{\partial u}) \prod_{j\in \partial u\setminus\{i\}} \nu_{j\to u}(x_i) \right\}$$

Variable to factor

$$\nu_{i\to u}(x_i) \stackrel{\text{def}}{=} \prod_{v\in \partial i\setminus\{u\}} \mu_{v\to i}(x_i)$$



$$\begin{aligned}
\mu_{c\to 3}(x_3) &= f_c(x_3)\\
\nu_{1\to a}(x_1) &= 1\\
\nu_{2\to a}(x_2) &= 1\\
\nu_{3\to b}(x_3) &= \mu_{c\to 3}(x_3) = f_c(x_3)\\
\mu_{a\to 0}(x_0) &= \max_{x_1,x_2} f_a(x_0,x_1,x_2) \times 1 \times 1\\
\mu_{b\to 0}(x_0) &= \max_{x_3} f_b(x_0,x_3) f_c(x_3)
\end{aligned}$$

$$\begin{aligned}
\nu_{0\to b}(x_0) &= \max_{x_1,x_2} \{f_a(x_0,x_1,x_2)\}\\
\nu_{0\to a}(x_0) &= \max_{x_3} \{f_b(x_0,x_3) f_c(x_3)\}\\
\mu_{b\to 3}(x_3) &= \max_{x_0} \left\{ f_b(x_0,x_3) \max_{x_1,x_2} \{f_a(x_0,x_1,x_2)\} \right\}\\
\mu_{a\to 1}(x_1) &= \max_{x_0,x_2} \left\{ f_a(x_0,x_1,x_2) \left(1 \times \max_{x_3} \{f_b(x_0,x_3) f_c(x_3)\} \right) \right\}\\
\mu_{a\to 2}(x_2) &= \max_{x_0,x_1} \left\{ f_a(x_0,x_1,x_2) \left(1 \times \max_{x_3} \{f_b(x_0,x_3) f_c(x_3)\} \right) \right\}
\end{aligned}$$

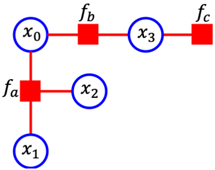
Example of Max-Product BP (2)

Most likely configuration:

$$\hat{x}_i = \arg\max_{x_i} \left\{ \prod_{v \in \partial i} \mu_{v \to i}(x_i) \right\} ,$$

$$p_{\max} = \max_{x_0} \{ \mu_{a \to 0}(x_0) \mu_{b \to 0}(x_0) \}$$

$$= \max_{x_0} \left\{ \left(\max_{x_1, x_2} f_a(x_0, x_1, x_2) \right) \left(\max_{x_3} f_b(x_0, x_3) f_c(x_3) \right) \right\} .$$



$$\mu_{c \to 3}(x_3) = f_c(x_3)$$
$$\nu_{1 \to a}(x_1) = 1$$
$$\nu_{2 \to a}(x_2) = 1$$
$$\nu_{3 \to b}(x_3) = \mu_{c \to 3}(x_3) = f_c(x_3)$$
$$\mu_{a \to 0}(x_0) = \max_{x_1, x_2} f_a(x_0, x_1, x_2) \times 1 \times 1$$
$$\mu_{b \to 0}(x_0) = \max_{x_3} f_b(x_0, x_3) f_c(x_3)$$

$$\nu_{0 \to b}(x_0) = \max_{x_1, x_2} \{ f_a(x_0, x_1, x_2) \}$$
$$\nu_{0 \to a}(x_0) = \max_{x_3} \{ f_b(x_0, x_3) f_c(x_3) \}$$
$$\mu_{b \to 3}(x_3) = \max_{x_0} \left\{ f_b(x_0, x_3) \max_{x_1, x_2} \{ f_a(x_0, x_1, x_2) \} \right\}$$
$$\mu_{a \to 1}(x_1) = \max_{x_0, x_2} \left\{ f_a(x_0, x_1, x_2) \left(1 \times \max_{x_3} \{ f_b(x_0, x_3) f_c(x_3) \} \right) \right\}$$
$$\mu_{a \to 2}(x_2) = \max_{x_0, x_1} \left\{ f_a(x_0, x_1, x_2) \left(1 \times \max_{x_3} \{ f_b(x_0, x_3) f_c(x_3) \} \right) \right\}$$

Max-Sum BP in Tree

- From factor f_u to variable x_i

$$\mu_{u\to i}(x_i) \stackrel{\text{def}}{=} \max_{x_{\partial u \setminus \{i\}}} \left\{ \log f_u(x_{\partial u}) + \sum_{j\in\partial u\setminus\{i\}} \nu_{j\to u}(x_j) \right\}$$

which is set to $\mu_{u\to i}(x_i) = \log f_u(x_i)$ if $\partial u = \{i\}$, i.e., leaf factor.

- From variable x_i to factor f_u

$$\nu_{i\to u}(x_i) \stackrel{\text{def}}{=} \sum_{v\in\partial i\setminus\{u\}} \mu_{v\to i}(x_i)$$

which is set to $\nu_{i\to u}(x_i) = 0$ if $\partial i = \{u\}$, i.e., leaf variable.

- Maximum probability $\log p_{\max} = \max_{x_i} \left\{ \sum_{v\in\partial i} \mu_{v\to i}(x_i) \right\}$, of which maximizer $\hat{x}_i$ forms the most likely configuration.

Table of Contents

Learning Graph Structure from Data

Suppose that we obtained an empirical distribution p of $X = (X_1, \ldots, X_N)$ from K samples: $\mathcal{D} = \{x^{(1)}, \ldots, x^{(K)}\}$ with $x^{(k)} = (x_1^{(k)}, ..., x_N^{(k)})$ for each $k \in [K]$.

- The empirical distribution is calculated as

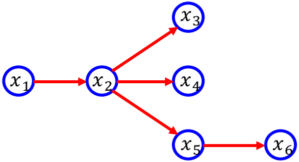
$$p(X = x) \stackrel{\text{def}}{=} \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}[x^{(k)} = x] \, .$$

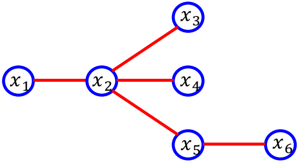
We want to build a simple graphical model or factorized joint distribution explaining/approximating the empirical distribution p.

A Simple Structure: Directed Tree

Given an empirical distribution p of $X = (X_1, \ldots, X_N)$, Chow-Liu algorithm constructs a Bayesian network of which factorization consists of second-order conditional and marginal distributions, e.g., a joint probability $p(x_1, x_2, x_3, x_4, x_5, x_6)$ might be approximated as

$$p(x_6 \mid x_5)p(x_5 \mid x_2)p(x_4 \mid x_2)p(x_3 \mid x_2)p(x_2 \mid x_1)p(x_1)$$





- A directed tree graph has no head-to-head topology since each child has one parent. Hence, the directed tree can be translated into an undirected tree graph.

Problem Formulation

A graph construction problem can be formulated as the minimization of distance between the empirical and (approximated) factorized distributions:

$$\underset{T:\text{tree}}{\text{minimize}} \quad \mathrm{KL}(p \| p_T) \stackrel{\text{def}}{=} \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{p_T(x)} \, .$$

- $\mathcal{X}$ is the set of all possible configuration of X.
- The minimization takes over every possible directed tree $T = (V, E)$ of N nodes.
- p_T is the distribution corresponding to Bayesian network T:

$$p_T(x) \stackrel{\text{def}}{=} \prod_{j \in V} p(x_j \mid x_{\mathrm{pa}(j)}) \, ,$$

  where for root j with no parent, $p(x_j \mid x_{\mathrm{pa}(j)}) = p(x_j \mid x_\emptyset) = p(x_j)$.

Factorization in Directed Tree

For a directed tree $T = (V, E)$, each node has one path from the root, or equivalently each child appears only once in conditional. Therefore, we can write

$$
\begin{aligned}
p_T(x) \stackrel{\text{def}}{=} \prod_{j \in V} p(x_j \mid x_{\mathrm{pa}(j)}) &= p(x_{\mathrm{root}}) \prod_{(i,j) \in E} p(x_j \mid x_i) \\
&= p(x_{\mathrm{root}}) \prod_{(i,j) \in E} \frac{p(x_j, x_i)}{p(x_i)} \\
&= p(x_{\mathrm{root}}) \prod_{(i,j) \in E} \frac{p(x_j, x_i) p(x_j)}{p(x_i) p(x_j)} \\
&= \left(\prod_{i \in V} p(x_i) \right) \left(\prod_{(i,j) \in E} \frac{p(x_i, x_j)}{p(x_i) p(x_j)} \right) .
\end{aligned}
$$

Finding The Best Approximation (1)

$$
\begin{aligned}
T^* &= \underset{T:\text{tree}}{\arg\min} \quad \mathrm{KL}(p \| p_T) \\
&= \underset{T:\text{tree}}{\arg\min} \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{p_T(x)} \\
&= \underset{T:\text{tree}}{\arg\max} \sum_{x \in \mathcal{X}} p(x) \log p_T(x) \\
&= \underset{T:\text{tree}}{\arg\max} \sum_{x \in \mathcal{X}} p(x) \log \left(\left(\prod_{i \in V} p(x_i) \right) \left(\prod_{(i,j) \in E} \frac{p(x_i, x_j)}{p(x_i) p(x_j)} \right) \right) \\
&= \underset{T:\text{tree}}{\arg\max} \sum_{x \in \mathcal{X}} \sum_{i \in V} p(x) \log p(x_i) + \sum_{x \in \mathcal{X}} \sum_{(i,j) \in E} p(x) \log \left(\frac{p(x_i, x_j)}{p(x_i) p(x_j)} \right)
\end{aligned}
$$

Finding The Best Approximation (2)

$$
\begin{aligned}
T^* &= \underset{T:\text{tree}}{\arg\min} \quad \mathrm{KL}(p \| p_T) \\
&= \underset{T:\text{tree}}{\arg\max} \sum_{x \in \mathcal{X}} \sum_{i \in V} p(x) \log p(x_i) + \sum_{x \in \mathcal{X}} \sum_{(i,j) \in E} p(x) \log \left(\frac{p(x_i, x_j)}{p(x_i) p(x_j)} \right) \\
&= \underset{T:\text{tree}}{\arg\max} \sum_{i \in V} \underbrace{\sum_{x_i} p(x_i) \log p(x_i)}_{\stackrel{\text{def}}{=} -H(X_i)} + \underbrace{\sum_{(i,j) \in E} \sum_{x_i, x_j} p(x_i, x_j) \log \left(\frac{p(x_i, x_j)}{p(x_i) p(x_j)} \right)}_{\stackrel{\text{def}}{=} I(X_i, X_j)},
\end{aligned}
$$

where $H(X)$ is the entropy of random variable X, and $I(X, Y)$ is the mutual information between random variables X and Y, which quantifies dependence between them: $X \perp\!\!\!\perp Y$, i.e., $p(X)p(Y) = p(X, Y)$ implies $I(X, Y) = 0$.

Finding The Best Approximation (3)

$$
\begin{aligned}
T^* &= \operatorname*{arg\,min}_{T\text{:tree}} \quad \mathrm{KL}(p \| p_T) \\
&= \operatorname*{arg\,max}_{T\text{:tree}} \sum_{i \in V} -H(X_i) + \sum_{(i,j) \in E} I(X_i, X_j) \\
&= \operatorname*{arg\,max}_{T\text{:tree}} \sum_{(i,j) \in E} I(X_i, X_j) \ ,
\end{aligned}
$$

which is the maximum weighted spanning tree (MWST) of undirected complete graph with weight $w(i,j) = I(X_i, X_j)$.

Chow-Liu Algorithm

$$T^* = \underset{T:\text{tree}}{\arg\min} \quad \mathrm{KL}(p \| p_T)$$

- Calculate $I(X_i, X_j)$ for all possible pair (i, j).
- Run Kruskal's greedy algorithm to find MWST:
  - Sort the pairs into decreasing order by weight $w(i, j)$. Let E be the set of edges comprising the maximum weight spanning tree. Set $E = \emptyset$ and add the first edge to E.
  - (*) Add the next edge to E if and only if it does not form a cycle in E.
  - If E has $N - 1$ edges, where N is the number of variable nodes, then stop and output $T^* = (V, E)$. Otherwise go to step (*).
- Pick an arbitrary node as a root and draw arrows aways.

Example

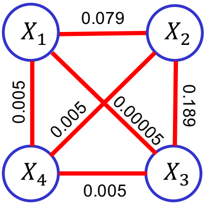
(x_1, x_2, x_3, x_4)	$p(x_1, x_2, x_3, x_4)$	$p(x_1)p(x_2)p(x_3)p(x_4)$
$(0, 0, 0, 0)$	0.10	0.046
$(0, 0, 0, 1)$	0.10	0.046
$(0, 0, 1, 0)$	0.05	0.056
$(0, 0, 1, 1)$	0.05	0.056
$(0, 1, 0, 0)$	0.00	0.056
$(0, 1, 0, 1)$	0.00	0.056
$(0, 1, 1, 0)$	0.10	0.068
$(0, 1, 1, 1)$	0.05	0.068
$(1, 0, 0, 0)$	0.05	0.056
$(1, 0, 0, 1)$	0.10	0.056
$(1, 0, 1, 0)$	0.00	0.068
$(1, 0, 1, 1)$	0.00	0.068
$(1, 1, 0, 0)$	0.05	0.068
$(1, 1, 0, 1)$	0.05	0.068
$(1, 1, 1, 0)$	0.15	0.083
$(1, 1, 1, 1)$	0.15	0.083

Example

(x_1, x_2, x_3, x_4)	$p(x_1, x_2, x_3, x_4)$
$(0,0,0,0)$	0.10
$(0,0,0,1)$	0.10
$(0,0,1,0)$	0.05
$(0,0,1,1)$	0.05
$(0,1,0,0)$	0.00
$(0,1,0,1)$	0.00
$(0,1,1,0)$	0.10
$(0,1,1,1)$	0.05
$(1,0,0,0)$	0.05
$(1,0,0,1)$	0.10
$(1,0,1,0)$	0.00
$(1,0,1,1)$	0.00
$(1,1,0,0)$	0.05
$(1,1,0,1)$	0.05
$(1,1,1,0)$	0.15
$(1,1,1,1)$	0.15

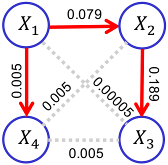
$$
\begin{aligned}
p(X_1 = 0) &= 0.1 + 0.1 + 0.05 + 0.05 + 0 + 0 + 0.1 + 0.05 = 0.45 \ , \\
p(X_2 = 0) &= 0.1 + 0.1 + 0.05 + 0.05 + 0.05 + 0.1 + 0 + 0 = 0.45 \ , \\
p(X_1 = 0, X_2 = 0) &= 0.1 + 0.1 + 0.05 + 0.05 = 0.3 \ , \\
p(X_1 = 0, X_2 = 1) &= 0.45 - 0.3 = 0.15 \ , \\
p(X_1 = 1, X_2 = 0) &= 0.05 + 0.1 + 0 + 0 = 0.15 \ , \\
p(X_1 = 1, X_2 = 1) &= 0.55 - 0.15 = 0.4 \ .
\end{aligned}
$$

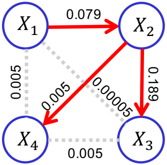
$$
\begin{aligned}
I(X_1, X_2) &= \sum_{x_1, x_2} p(x_1, x_2) \log \frac{p(x_1, x_2)}{p(x_1)p(x_2)} \\
&= 0.3 \log \frac{0.3}{0.45 * 0.45} + 0.15 \log \frac{0.15}{0.45 * 0.55} \\
&\quad + 0.15 \log \frac{0.15}{0.55 * 0.45} + 0.4 \log \frac{0.4}{0.55 * 0.55} = 0.0794\ldots
\end{aligned}
$$

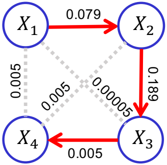


Example

(x_1, x_2, x_3, x_4)	$p(x_1, x_2, x_3, x_4)$	$p(x_1)p(x_2)p(x_3)p(x_4)$	$p(x_1)p(x_2\|x_1)p(x_3\|x_2)p(x_4\|x_1)$
(0, 0, 0, 0)	0.10	0.046	0.130
(0, 0, 0, 1)	0.10	0.046	0.104
(0, 0, 1, 0)	0.05	0.056	0.037
(0, 0, 1, 1)	0.05	0.056	0.030
(0, 1, 0, 0)	0.00	0.056	0.015
(0, 1, 0, 1)	0.00	0.056	0.012
(0, 1, 1, 0)	0.10	0.068	0.068
(0, 1, 1, 1)	0.05	0.068	0.054
(1, 0, 0, 0)	0.05	0.056	0.053
(1, 0, 0, 1)	0.10	0.056	0.064
(1, 0, 1, 0)	0.00	0.068	0.015
(1, 0, 1, 1)	0.00	0.068	0.018
(1, 1, 0, 0)	0.05	0.068	0.033
(1, 1, 0, 1)	0.05	0.068	0.040
(1, 1, 1, 0)	0.15	0.083	0.149
(1, 1, 1, 1)	0.15	0.083	0.178







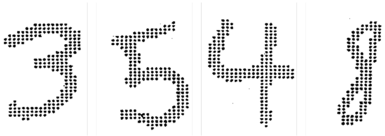
Application to Pattern Recognition (1)

Suppose that we have a labeled data of $K = 19,000+$ scanned images of numeral $(0-9)$, where each image $x^{(k)} = (x_1^{(k)}, \ldots, x_{96}^{(k)})$ contains 96 (12 x 8) binary dots for a handwriting of single numeral $\ell^{(k)}$.

- We want to learn patterns of handwritings, and build a pattern recognition system classifying new handwriting inputs.
- Let p_ℓ be the prior distribution of $\ell \in \{0, \ldots, 9\}$. For a new (unlabeled) input x^{new}, a reasonable decision rule may take

$$\arg\max_{\ell \in \{0,\ldots,9\}} \; p_\ell \times p(x^{\text{new}} \mid \ell^{(new)} = \ell) \, .$$

- But, we cannot have dataset for all possible images (2^{96} many possibilities $\gg 19,000+$), i.e., we need to learn pattern from observation to conjecture the true label unseen one. Any idea? Chow-Liu algorithm!



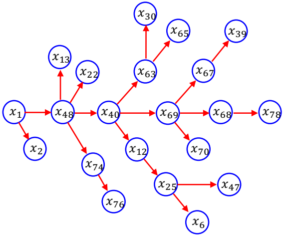
Application to Pattern Recognition (2)

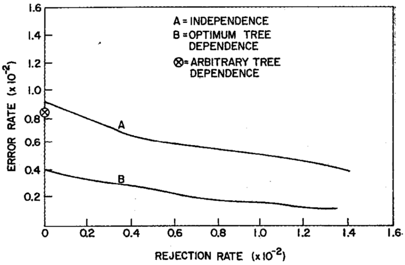
Suppose that we have a labeled data of $K = 19,000+$ scanned images of numeral $(0-9)$, where each image $x^{(k)} = (x_1^{(k)}, ..., x_{96}^{(k)})$ contains 96 (12 x 8) binary dots for a handwriting of single numeral $\ell^{(k)}$.

- For each $\ell \in \{0, ..., 9\}$, to approximate $p(x^{\text{new}} \mid \ell^{(new)} = \ell)$, we run Chow-Liu algorithm for empirical conditional distribution of $x^{(k)}$ given $\ell^{(k)} = \ell$, and make decisions based on the output p_{T_ℓ}:

$$\arg\max_{\ell \in \{0,\dots,9\}} \; p_\ell \times \underbrace{p_{T_\ell}(x^{\text{new}})}_{\approx \, p(x^{\text{new}} \mid \ell^{new} = \ell)} \quad ,$$

where T_ℓ is the output tree graphical model, e.g., a part of $T_{\ell=4}$,





Summary

- Statistical graphical model visualizes dependence among variables by translating the factorized joint probability into a graph (directed/undirected/factor).
- Efficient algorithms for machine learning, e.g., sum-product/max-product BP for marginalization/maximization, can be developed from the graphical model.
- Graphical model can be constructed from data, e.g., Chow-Liu algorithm.

Research Problems

- Analysis of loopy BP
  - Sanghavi, Sujay, Dmitry Malioutov, and Alan S. Willsky. "Linear programming analysis of loopy belief propagation for weighted matching." Advances in neural information processing systems. 2008.
  - Mossel, E., Neeman, J., and Sly, A. Belief propagation, robust reconstruction and optimal recovery of block models. In Proceedings of COLT, 2014.
- BP for distributed algorithm
  - Jonathan S Yedidia, William T Freeman, and Yair Weiss. "Constructing free-energy approximations and generalized belief propagation algorithms." Information Theory, IEEE Transactions on, 51(7):2282–2312, 2005.
  - David Gamarnik, Devavrat Shah, and Yehua Wei. "Belief propagation for min-cost network flow: Convergence and correctness." Operations Research, 60(2):410–428, 2012.
  - Park, S. and Shin, J. "Max-product belief propagation for linear programming: applications to combinatorial optimization." In Proceedings of UAI, 2015.
- Graph construction with hidden variables
  - Bresler, Guy. "Efficiently learning Ising models on arbitrary graphs." Proceedings of the forty-seventh annual ACM symposium on Theory of computing. ACM, 2015.
- Graphical neural network and graphical models
  - Satorras, Victor Garcia, and Max Welling. "Neural Enhanced Belief Propagation on Factor Graphs." arXiv preprint arXiv:2003.01998 (2020).