Chow-Liu Trees

Generative AI Models — Lecture 5

9th May 2025

Thomas Krak (slides adapted from Gennaro Gala)
Uncertainty in Artificial Intelligence

Idea: structure learning as discrete optimization

- Let $\boldsymbol{X}$ be a set of RVs and $\mathcal{D} = \{\boldsymbol{x}^n\}_{n=1}^N$ be i.i.d. data
- Let $[\mathcal{G}]$ be some family of DAGs over $\boldsymbol{X}$
- Define a suitable **score** $\mathcal{S}(\mathcal{G}, \mathcal{D})$
- Find $\mathcal{G}^* = \arg\max_{\mathcal{G} \in [\mathcal{G}]} \mathcal{S}(\mathcal{G}, \mathcal{D})$

- $[\mathcal{G}]$ is the **set of all directed trees** $[\mathcal{T}]$ over $\boldsymbol{X}$
- **Directed tree**: Every RV has at most one parent
- Score $\mathcal{S}(\mathcal{G}, \mathcal{D}) = \max_{\Theta} \mathcal{L}(\mathcal{G}, \Theta, \mathcal{D})$, where
  - Θ are all BN parameters for $\mathcal{G}$ (for categorical CPDs)
  - $\mathcal{L}(\mathcal{G}, \Theta, \mathcal{D})$ is the log-likelihood
- Thus, tree $\mathcal{G}$ is "better" than $\mathcal{G}'$ if the log-likelihood of $\mathcal{G}$ is higher than the log-likelihood of $\mathcal{G}'$ (when equipping them with their ML parameters):

$$\mathcal{G}^* = \arg\max_{\mathcal{G} \in [\mathcal{T}]} \underbrace{\left(\max_{\Theta} \mathcal{L}(\mathcal{G}, \Theta, \mathcal{D}) \right)}_{\mathcal{S}}$$

- Remarkable: **Poly-time Algorithm!**

Algorithm 3 VE_PR1($\mathcal{N}$, $\mathbf{Q}$, π)

input:

$\mathcal{N}$: Bayesian network

$\mathbf{Q}$: variables in network $\mathcal{N}$

π: ordering of network variables not in $\mathbf{Q}$

output: the prior marginal $\Pr(\mathbf{Q})$

main:

1: $\mathcal{S} \leftarrow$ CPTs of network $\mathcal{N}$
2: **for** $i = 1$ to length of order π **do**
3: $\quad f \leftarrow \prod_k f_k$, where f_k belongs to $\mathcal{S}$ and mentions variable $\pi(i)$
4: $\quad f_i \leftarrow \sum_{\pi(i)} f$
5: $\quad$ replace all factors f_k in $\mathcal{S}$ by factor f_i
6: **end for**
7: **return** $\prod_{f \in \mathcal{S}} f$

Reading Assignment 2

Approximating Discrete Probability Distributions with Dependence Trees

C. K. CHOW, SENIOR MEMBER, IEEE, AND C. N. LIU, MEMBER, IEEE

Overview

Learning Chow-Liu Trees

Inference in Tree-shaped BNs

Ancestral Sampling

Learning Chow-Liu Trees

(Kullback–Leibler divergence) Let p and q be probability distributions over the same state space $\mathcal{X}$. The Kullback-Leibler divergence between p and q is defined as:

$$\mathbb{KL}(p||q) = \sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \geq 0 \tag{1}$$

(Kullback–Leibler divergence) Let p and q be probability distributions over the same state space $\boldsymbol{\mathcal{X}}$. The Kullback-Leibler divergence between p and q is defined as:

$$\mathbb{KL}(p||q) = \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \geq 0 \tag{1}$$

In other words, it is the expectation of the logarithmic difference between the distributions p and q, where the expectation is taken using the distribution p, i.e.:

$$\mathbb{KL}(p||q) = \mathbb{E}_{\boldsymbol{x} \sim p}\left[\log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right] = \mathbb{E}_{\boldsymbol{x} \sim p}\left[\log p(\boldsymbol{x}) - \log q(\boldsymbol{x})\right] \tag{2}$$

(Kullback–Leibler divergence) Let p and q be probability distributions over the same state space $\mathcal{X}$. The Kullback-Leibler divergence between p and q is defined as:

$$\mathbb{KL}(p||q) = \sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \geq 0 \tag{1}$$

In other words, it is the expectation of the logarithmic difference between the distributions p and q, where the expectation is taken using the distribution p, i.e.:

$$\mathbb{KL}(p||q) = \mathbb{E}_{\boldsymbol{x} \sim p}\left[\log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right] = \mathbb{E}_{\boldsymbol{x} \sim p}\left[\log p(\boldsymbol{x}) - \log q(\boldsymbol{x})\right] \tag{2}$$

Note that, in general, $\mathbb{KL}(p||q) \neq \mathbb{KL}(q||p)$ and that $\mathbb{KL}(p||q) = 0$ iff $p = q$.

(Mutual Information) Given two jointly discrete RVs X and Y with joint distribution p_{XY} and marginal distributions p_X and p_Y, the mutual information $\mathrm{MI}(X;Y)$ between X and Y is:

$$\mathrm{MI}(X;Y) = \mathbb{KL}(p_{XY}||p_X p_Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)} \tag{3}$$

(Mutual Information) Given two jointly discrete RVs X and Y with joint distribution p_{XY} and marginal distributions p_X and p_Y, the mutual information $\mathrm{MI}(X;Y)$ between X and Y is:

$$\mathrm{MI}(X;Y) = \mathbb{KL}(p_{XY}||p_X p_Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)} \tag{3}$$

- The mutual information of two RVs is a measure of the mutual dependence

(Mutual Information) Given two jointly discrete RVs X and Y with joint distribution p_{XY} and marginal distributions p_X and p_Y, the mutual information $\mathrm{MI}(X;Y)$ between X and Y is:

$$\mathrm{MI}(X;Y) = \mathbb{KL}(p_{XY}||p_X p_Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)} \tag{3}$$

- The mutual information of two RVs is a measure of the mutual dependence
- Note that, MI is measured in *nats* (natural unit of information) when the natural logarithm is used.

$p_{XY}(X,Y)$	$x=0$	$x=1$	$p_Y(Y)$
$y=0$	0.1	0.3	0.4
$y=1$	0.2	0.4	0.6
$p_X(X)$	0.3	0.7	

$p_{XY}(X,Y)$	$x=0$	$x=1$	$p_Y(Y)$
$y=0$	0.1	0.3	0.4
$y=1$	0.2	0.4	0.6
$p_X(X)$	0.3	0.7	

$$
\begin{aligned}
\mathrm{MI}(X;Y) &= \mathbb{KL}(p_{XY}||p_X p_Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)} = \\
&= 0.1\log\frac{0.1}{0.3\cdot 0.4} + 0.3\log\frac{0.3}{0.7\cdot 0.4} + 0.2\log\frac{0.2}{0.3\cdot 0.6} + 0.4\log\frac{0.4}{0.7\cdot 0.6} \\
&\approx 0.004 \text{ nats}
\end{aligned}
$$

$p_{XY}(X, Y)$	$x = 0$	$x = 1$	$p_Y(Y)$
$y = 0$	0.08	0.32	0.4
$y = 1$	0.12	0.48	0.6
$p_X(X)$	0.2	0.8	

$p_{XY}(X,Y)$	$x=0$	$x=1$	$p_Y(Y)$
$y=0$	0.08	0.32	0.4
$y=1$	0.12	0.48	0.6
$p_X(X)$	0.2	0.8	

$$
\begin{aligned}
\mathrm{MI}(X;Y) &= \mathbb{KL}(p_{XY}||p_X p_Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)} = \\
&= 0.08\log\frac{0.08}{0.2\cdot 0.4} + 0.32\log\frac{0.32}{0.8\cdot 0.4} + 0.12\log\frac{0.12}{0.2\cdot 0.6} + 0.48\log\frac{0.48}{0.8\cdot 0.6} \\
&= 0 \text{ nats}
\end{aligned}
$$

Bayesian Networks

A *Bayesian Network* (BN) over RVs $\boldsymbol{X} = (X_i)_{i=1}^d$ is a pair $(\mathcal{G}, \mathcal{P})$, where:

- $\mathcal{G}$ is a DAG which has RVs $\boldsymbol{X}$ as nodes;
- $\mathcal{P}$ is a collection of distributions $p(X_i|\mathbf{pa}(X_i))$;

and where:

$$p(\boldsymbol{X}) = \prod_{i=1}^{d} p(X_i|\mathbf{pa}(X_i)).$$

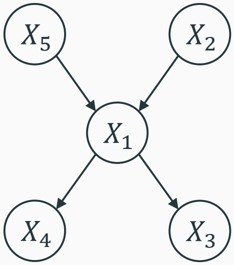
Bayesian Networks

A *Bayesian Network* (BN) over RVs $\boldsymbol{X} = (X_i)_{i=1}^d$ is a pair $(\mathcal{G}, \mathcal{P})$, where:

- $\mathcal{G}$ is a DAG which has RVs $\boldsymbol{X}$ as nodes;
- $\mathcal{P}$ is a collection of distributions $p(X_i|\mathbf{pa}(X_i))$;

and where:

$$p(\boldsymbol{X}) = \prod_{i=1}^{d} p(X_i|\mathbf{pa}(X_i)).$$



$$p(\boldsymbol{X}) = p(X_1|X_2, X_5)p(X_2)p(X_3|X_1)p(X_4|X_1)p(X_5)$$

Tree-shaped Bayesian Networks

A *tree-shaped* BN over RVs $\boldsymbol{X} = (X_i)_{i=1}^d$ is a pair $(\mathcal{T}, \mathcal{P})$, where:

- $\mathcal{T}$ is a directed tree which has RVs $\boldsymbol{X}$ as nodes;
- $\mathcal{P}$ is a collection of distributions $p(X_i | X_{\tau(i)})$, where $X_{\tau(i)}$ is the parent of X_i in $\mathcal{T}$;

and where:

$$p(\boldsymbol{X}) = \prod_{i=1}^{d} p(X_i | X_{\tau(i)}).$$

If X_i is the root of $\mathcal{T}$ then $\tau(i) = 0$ and $p(X_i | X_0) = p(X_i)$.

Tree-shaped Bayesian Networks

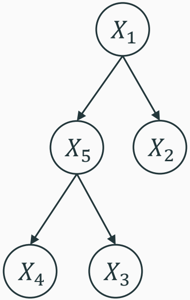
A *tree-shaped* BN over RVs $\boldsymbol{X} = (X_i)_{i=1}^d$ is a pair $(\mathcal{T}, \mathcal{P})$, where:

- $\mathcal{T}$ is a directed tree which has RVs $\boldsymbol{X}$ as nodes;
- $\mathcal{P}$ is a collection of distributions $p(X_i | X_{\tau(i)})$, where $X_{\tau(i)}$ is the parent of X_i in $\mathcal{T}$;

and where:

$$p(\boldsymbol{X}) = \prod_{i=1}^{d} p(X_i | X_{\tau(i)}).$$

If X_i is the root of $\mathcal{T}$ then $\tau(i) = 0$ and $p(X_i | X_0) = p(X_i)$.



$$p(\boldsymbol{X}) = p(X_1)p(X_2|X_1)p(X_3|X_5)p(X_4|X_5)p(X_5|X_1)$$

Problem

- We are given a dataset $\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ drawn from an unknown distribution $p^*(\boldsymbol{X})$

Problem

- We are given a dataset $\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ drawn from an unknown distribution $p^*(\boldsymbol{X})$
- We want to learn the "best" tree-shaped BN $(\mathcal{T}, \mathcal{P})$ from $\mathcal{D}$

Problem

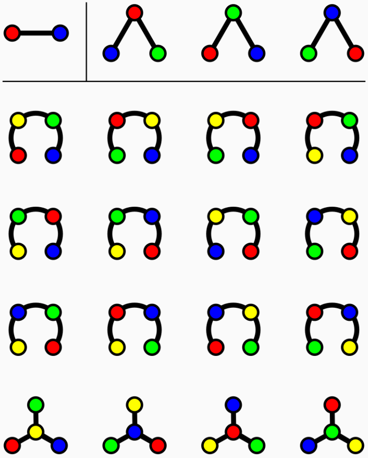
- We are given a dataset $\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ drawn from an unknown distribution $p^*(\boldsymbol{X})$
- We want to learn the "best" tree-shaped BN $(\mathcal{T}, \mathcal{P})$ from $\mathcal{D}$
- In other words, we want to find the best tree-based approximation $p(\boldsymbol{X}) = \prod_{i=1}^{d} p^*(X_i | X_{\tau(i)})$ of $p^*(\boldsymbol{X})$

How many possible trees?

- Cayley's formula is a result in graph theory named after Arthur Cayley. It states that for every positive integer d, the number of trees on d labeled vertices is d^{d-2}
- The number of possible trees for any moderate value of d is so enormous as to exlude any approach of exhaustive search

How many possible trees?

- Cayley's formula is a result in graph theory named after Arthur Cayley. It states that for every positive integer d, the number of trees on d labeled vertices is d^{d-2}
- The number of possible trees for any moderate value of d is so enormous as to exlude any approach of exhaustive search



Chow-Liu Algorithm - The Proof \1

We want to find $\mathcal{T}$ s.t. its induced probability distribution $p(\boldsymbol{X}) = \prod_{i=1}^{d} p^*(X_i|X_{\tau(i)})$ is as close as possible to the true unknown distribution $p^*(\boldsymbol{X})$.

Chow-Liu Algorithm - The Proof \1

We want to find $\mathcal{T}$ s.t. its induced probability distribution $p(\boldsymbol{X}) = \prod_{i=1}^{d} p^*(X_i | X_{\tau(i)})$ is as close as possible to the true unknown distribution $p^*(\boldsymbol{X})$.

$$\mathbb{KL}(p^* || p) = \mathbb{E}_{\boldsymbol{x} \sim p^*} \left[\log p^*(\boldsymbol{x}) - \log p(\boldsymbol{x})\right]$$

Chow-Liu Algorithm - The Proof \1

We want to find $\mathcal{T}$ s.t. its induced probability distribution $p(\boldsymbol{X}) = \prod_{i=1}^{d} p^*(X_i | X_{\tau(i)})$ is as close as possible to the true unknown distribution $p^*(\boldsymbol{X})$.

$$\begin{aligned}\mathbb{KL}(p^* || p) &= \mathbb{E}_{\boldsymbol{x} \sim p^*} \left[\log p^*(\boldsymbol{x}) - \log p(\boldsymbol{x})\right] \\ &= \mathbb{E}_{\boldsymbol{x} \sim p^*} \left[\log p^*(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x} \sim p^*} \left[\log p(\boldsymbol{x})\right]\end{aligned}$$

Chow-Liu Algorithm - The Proof \1

We want to find $\mathcal{T}$ s.t. its induced probability distribution $p(\boldsymbol{X}) = \prod_{i=1}^{d} p^*(X_i|X_{\tau(i)})$ is as close as possible to the true unknown distribution $p^*(\boldsymbol{X})$.

$$
\begin{aligned}
\mathbb{KL}(p^*||p) &= \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log p^*(\boldsymbol{x}) - \log p(\boldsymbol{x})\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log p^*(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log p(\boldsymbol{x})\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log p^*(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d} \log p^*(\boldsymbol{x}_i|\boldsymbol{x}_{\tau(i)})\right]
\end{aligned}
$$

Chow-Liu Algorithm - The Proof \1

We want to find $\mathcal{T}$ s.t. its induced probability distribution $p(\boldsymbol{X}) = \prod_{i=1}^{d} p^*(X_i|X_{\tau(i)})$ is as close as possible to the true unknown distribution $p^*(\boldsymbol{X})$.

$$\begin{aligned}
\mathbb{KL}(p^*||p) &= \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log p^*(\boldsymbol{x}) - \log p(\boldsymbol{x})\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log p^*(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log p(\boldsymbol{x})\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log p^*(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d} \log p^*(\boldsymbol{x}_i|\boldsymbol{x}_{\tau(i)})\right]
\end{aligned}$$

Since $\mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\log p^*(\boldsymbol{x})\right]$ is independent of $\mathcal{T}$, only the second quantity matters.

Chow-Liu Algorithm - The Proof \2

$$\mathbb{E}_{\boldsymbol{x} \sim p^*} \left[\sum_{i=1}^{d} \log p^*(\boldsymbol{x}_i | \boldsymbol{x}_{\tau(i)}) \right]$$

Chow-Liu Algorithm - The Proof \2

$$\mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log p^*(\boldsymbol{x}_i|\boldsymbol{x}_{\tau(i)})\right] = \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log \frac{p^*(\boldsymbol{x}_i|\boldsymbol{x}_{\tau(i)})p^*(\boldsymbol{x}_{\tau(i)})p^*(\boldsymbol{x}_i)}{p^*(\boldsymbol{x}_i)p^*(\boldsymbol{x}_{\tau(i)})}\right] =$$

Chow-Liu Algorithm - The Proof \2

$$\mathbb{E}_{\mathbf{x}\sim p^*}\left[\sum_{i=1}^{d}\log p^*(\mathbf{x}_i|\mathbf{x}_{\tau(i)})\right] = \mathbb{E}_{\mathbf{x}\sim p^*}\left[\sum_{i=1}^{d}\log \frac{p^*(\mathbf{x}_i|\mathbf{x}_{\tau(i)})p^*(\mathbf{x}_{\tau(i)})p^*(\mathbf{x}_i)}{p^*(\mathbf{x}_i)p^*(\mathbf{x}_{\tau(i)})}\right] =$$

$$= \mathbb{E}_{\mathbf{x}\sim p^*}\left[\sum_{i=1}^{d}\log \frac{p^*(\mathbf{x}_i, \mathbf{x}_{\tau(i)})}{p^*(\mathbf{x}_i)p^*(\mathbf{x}_{\tau(i)})}\right] + \mathbb{E}_{\mathbf{x}\sim p^*}\left[\sum_{i=1}^{d}\log p^*(\mathbf{x}_i)\right]$$

Chow-Liu Algorithm - The Proof \2

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log p^*(\boldsymbol{x}_i|\boldsymbol{x}_{\tau(i)})\right] &= \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log \frac{p^*(\boldsymbol{x}_i|\boldsymbol{x}_{\tau(i)})p^*(\boldsymbol{x}_{\tau(i)})p^*(\boldsymbol{x}_i)}{p^*(\boldsymbol{x}_i)p^*(\boldsymbol{x}_{\tau(i)})}\right] = \\
&= \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log \frac{p^*(\boldsymbol{x}_i,\boldsymbol{x}_{\tau(i)})}{p^*(\boldsymbol{x}_i)p^*(\boldsymbol{x}_{\tau(i)})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log p^*(\boldsymbol{x}_i)\right] \\
&= \sum_{i=1}^{d}\mathrm{MI}(X_i, X_{\tau(i)}) + \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log p^*(\boldsymbol{x}_i)\right]
\end{aligned}
$$

Chow-Liu Algorithm - The Proof \2

$$\mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log p^*(\boldsymbol{x}_i|\boldsymbol{x}_{\tau(i)})\right] = \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log \frac{p^*(\boldsymbol{x}_i|\boldsymbol{x}_{\tau(i)})p^*(\boldsymbol{x}_{\tau(i)})p^*(\boldsymbol{x}_i)}{p^*(\boldsymbol{x}_i)p^*(\boldsymbol{x}_{\tau(i)})}\right] =$$

$$= \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log \frac{p^*(\boldsymbol{x}_i,\boldsymbol{x}_{\tau(i)})}{p^*(\boldsymbol{x}_i)p^*(\boldsymbol{x}_{\tau(i)})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log p^*(\boldsymbol{x}_i)\right]$$

$$= \sum_{i=1}^{d}\mathrm{MI}(X_i, X_{\tau(i)}) + \mathbb{E}_{\boldsymbol{x}\sim p^*}\left[\sum_{i=1}^{d}\log p^*(\boldsymbol{x}_i)\right]$$

Therefore, minimising $\mathbb{KL}(p^*||p)$ is equivalent to maximizing $\sum_{i=1}^{d}\mathrm{MI}(X_i, X_{\tau(i)})$ over all possible trees.

Maximum spanning tree

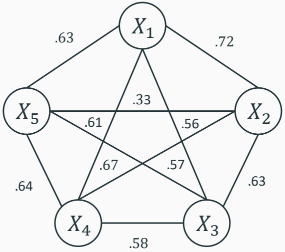
Let MI be the Mutual Information matrix of $\boldsymbol{X} = (X_i)_{i=1}^5$.

$$MI = \begin{bmatrix} & .72 & .56 & .61 & .63 \\ .72 & & .63 & .57 & .33 \\ .56 & .63 & & .58 & .67 \\ .61 & .57 & .58 & & .64 \\ .63 & .33 & .67 & .64 & \end{bmatrix}$$

Maximum spanning tree

Let MI be the Mutual Information matrix of $\boldsymbol{X} = (X_i)_{i=1}^{5}$.

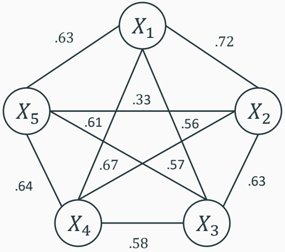
$$MI = \begin{bmatrix} & .72 & .56 & .61 & .63 \\ .72 & & .63 & .57 & .33 \\ .56 & .63 & & .58 & .67 \\ .61 & .57 & .58 & & .64 \\ .63 & .33 & .67 & .64 & \end{bmatrix}$$

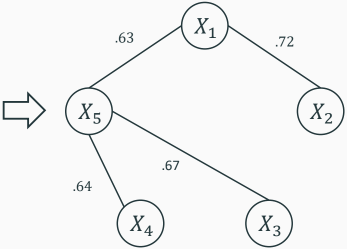


Maximum spanning tree

Let MI be the Mutual Information matrix of $\boldsymbol{X} = (X_i)_{i=1}^{5}$.

$$MI = \begin{bmatrix} & .72 & .56 & .61 & .63 \\ .72 & & .63 & .57 & .33 \\ .56 & .63 & & .58 & .67 \\ .61 & .57 & .58 & & .64 \\ .63 & .33 & .67 & .64 & \end{bmatrix}$$

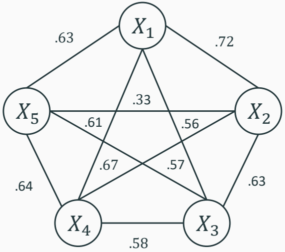


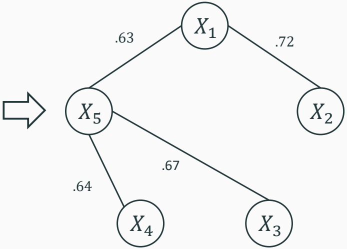


Maximum spanning tree

Let MI be the Mutual Information matrix of $\boldsymbol{X} = (X_i)_{i=1}^5$.

$$MI = \begin{bmatrix} & .72 & .56 & .61 & .63 \\ .72 & & .63 & .57 & .33 \\ .56 & .63 & & .58 & .67 \\ .61 & .57 & .58 & & .64 \\ .63 & .33 & .67 & .64 & \end{bmatrix}$$





- *A* **maximum spanning tree** *is a subset of the edges of a connected undirected graph that connects all the vertices together, without any cycles and with the maximum possible total edge weight*
- Kruskal's algorithm finds the maximum spanning tree in polynomial time

Orienting the Tree

Recall: minimising $\mathbb{KL}(p^*||p)$ is equivalent to maximizing $\sum_{i=1}^{d} \mathrm{MI}(X_i, X_{\tau(i)})$.

Mutual information is symmetric: $\mathrm{MI}(X_i, X_{\tau(i)}) = \mathrm{MI}(X_{\tau(i)}, X_i)$

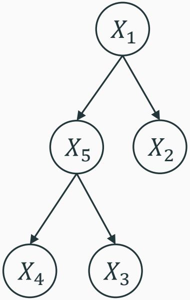
So direction of the arcs does not impact $\mathbb{KL}(p^*||p)$!

To orient the undirected maximum spanning tree:

- Choose any node as the root;
- Orient all edges to point away from the root

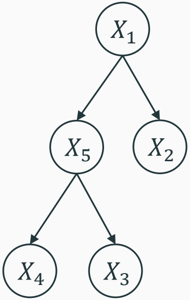
Parameter estimation

A CLT $(\mathcal{T}, \mathcal{P})$ encoding $p(\boldsymbol{X}) = p(X_1)p(X_2|X_1)p(X_3|X_5)p(X_4|X_5)p(X_5|X_1)$.



Parameter estimation

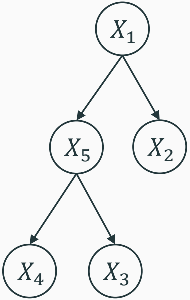
A CLT $(\mathcal{T}, \mathcal{P})$ encoding $p(\boldsymbol{X}) = p(X_1)p(X_2|X_1)p(X_3|X_5)p(X_4|X_5)p(X_5|X_1)$.



$$p(Y = y|Z = z) = \frac{p(Y = y, Z = z)}{p(Z = z)}$$

Parameter estimation

A CLT $(\mathcal{T}, \mathcal{P})$ encoding $p(\boldsymbol{X}) = p(X_1)p(X_2|X_1)p(X_3|X_5)p(X_4|X_5)p(X_5|X_1)$.

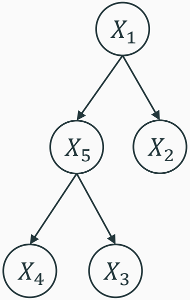


$$p(Y = y | Z = z) = \frac{p(Y = y, Z = z)}{p(Z = z)}$$

$$p(Y = y, Z = z) = \frac{\sum_{\boldsymbol{x} \in \mathcal{D}} \mathbb{1}[\boldsymbol{x}[Y] = y, \boldsymbol{x}[Z] = z]}{|\mathcal{D}|}$$

Parameter estimation

A CLT $(\mathcal{T}, \mathcal{P})$ encoding $p(\boldsymbol{X}) = p(X_1)p(X_2|X_1)p(X_3|X_5)p(X_4|X_5)p(X_5|X_1)$.



$$p(Y = y|Z = z) = \frac{p(Y = y, Z = z)}{p(Z = z)}$$

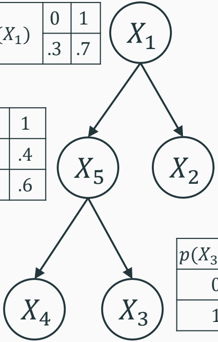
$$p(Y = y, Z = z) = \frac{\sum_{\boldsymbol{x} \in \mathcal{D}} \mathbb{1}[\boldsymbol{x}[Y] = y, \boldsymbol{x}[Z] = z]}{|\mathcal{D}|}$$

$$p(Z = z) = \frac{\sum_{\boldsymbol{x} \in \mathcal{D}} \mathbb{1}[\boldsymbol{x}[Z] = z]}{|\mathcal{D}|}$$

Parameter estimation

A CLT $(\mathcal{T}, \mathcal{P})$ encoding $p(\boldsymbol{X}) = p(X_1)p(X_2|X_1)p(X_3|X_5)p(X_4|X_5)p(X_5|X_1)$.

$p(X_1)$	0	1
	.3	.7



$p(X_5\|X_1)$	0	1
0	.9	.4
1	.1	.6

$p(X_2\|X_1)$	0	1
0	.2	.6
1	.8	.4

$p(X_4\|X_5)$	0	1
0	.8	.5
1	.2	.5

$p(X_3\|X_5)$	0	1
0	.4	.1
1	.6	.9

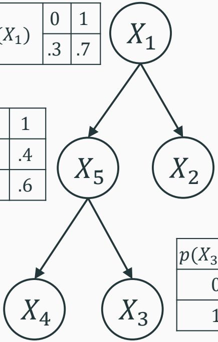
$$p(Y = y | Z = z) = \frac{p(Y = y, Z = z)}{p(Z = z)}$$

$$p(Y = y, Z = z) = \frac{\sum_{\boldsymbol{x} \in \mathcal{D}} \mathbb{1}[\boldsymbol{x}[Y] = y, \boldsymbol{x}[Z] = z]}{|\mathcal{D}|}$$

$$p(Z = z) = \frac{\sum_{\boldsymbol{x} \in \mathcal{D}} \mathbb{1}[\boldsymbol{x}[Z] = z]}{|\mathcal{D}|}$$

Parameter estimation

A CLT $(\mathcal{T}, \mathcal{P})$ encoding $p(\boldsymbol{X}) = p(X_1)p(X_2|X_1)p(X_3|X_5)p(X_4|X_5)p(X_5|X_1)$.



$p(X_1)$	0	1
	.3	.7

$p(X_5\|X_1)$	0	1
0	.9	.4
1	.1	.6

$p(X_2\|X_1)$	0	1
0	.2	.6
1	.8	.4

$p(X_4\|X_5)$	0	1
0	.8	.5
1	.2	.5

$p(X_3\|X_5)$	0	1
0	.4	.1
1	.6	.9

$$p(Y = y|Z = z) = \frac{p(Y = y, Z = z)}{p(Z = z)}$$

$$p(Y = y, Z = z) = \frac{\alpha + \sum_{\boldsymbol{x} \in \mathcal{D}} \mathbb{1}[\boldsymbol{x}[Y] = y, \boldsymbol{x}[Z] = z]}{4\alpha + |\mathcal{D}|}$$

$$p(Z = z) = \frac{2\alpha + \sum_{\boldsymbol{x} \in \mathcal{D}} \mathbb{1}[\boldsymbol{x}[Z] = z]}{4\alpha + |\mathcal{D}|}$$

where $\alpha > 0$ is a smoothing parameter for the Laplace's correction. Usually $\alpha = 0.01$.

Chow-Liu Algorithm

Algorithm 1 $\textsc{Learn-CLT}(\mathcal{D}, \alpha)$

Input: A set of samples $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$ over RVs $\mathbf{X}$ and a smoothing parameter α

Output: A CLT $(\mathcal{T}, \mathcal{P})$ over RVs $\mathbf{X}$

1: $MI \leftarrow \text{estimateMI}(\mathcal{D}, \alpha)$
2: $T \leftarrow \text{maximumSpanningTree}(MI)$
3: $\mathcal{T} \leftarrow \text{directedTree}(T)$
4: $\mathcal{P} \leftarrow \text{estimatePMFs}(\mathcal{T}, \mathcal{D}, \alpha)$
5: **return** $\langle \mathcal{T}, \mathcal{P} \rangle$

Chow-Liu Trees:

- Maximum-likelihood fit to given data over space of tree-shaped BNs
- Based on maximum-spanning tree for pairwise mutual information
- Runs in polynomial time using e.g. Kruskal's or Prim's algorithm

Inference in Tree-shaped BNs

Suppose we have a tree-shaped BN $(\mathcal{T}, \mathcal{P})$

- For instance, a CLT

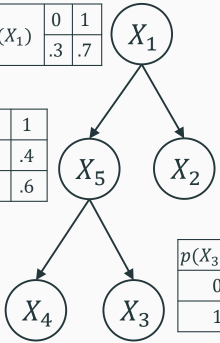
We now want to perform inference with it:

- Marginal inference
- Most Probably Explanation

How can we do this efficiently?

Inference in BNs

Consider a CLT $(\mathcal{T}, \mathcal{P})$ encoding $p(\boldsymbol{X}) = p(X_1)p(X_2|X_1)p(X_3|X_5)p(X_4|X_5)p(X_5|X_1)$.



$p(X_1)$	0	1
	.3	.7

$p(X_5\|X_1)$	0	1
0	.9	.4
1	.1	.6

$p(X_2\|X_1)$	0	1
0	.2	.6
1	.8	.4

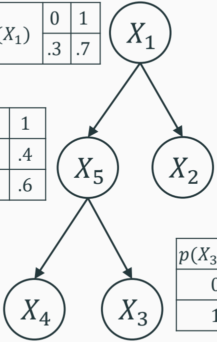
$p(X_4\|X_5)$	0	1
0	.8	.5
1	.2	.5

$p(X_3\|X_5)$	0	1
0	.4	.1
1	.6	.9

Inference in BNs

Consider a CLT $(\mathcal{T}, \mathcal{P})$ encoding $p(\boldsymbol{X}) = p(X_1)p(X_2|X_1)p(X_3|X_5)p(X_4|X_5)p(X_5|X_1)$.

$p(X_1)$	0	1
	.3	.7



$p(X_5 \mid X_1)$	0	1
0	.9	.4
1	.1	.6

$p(X_2 \mid X_1)$	0	1
0	.2	.6
1	.8	.4

$p(X_4 \mid X_5)$	0	1
0	.8	.5
1	.2	.5

$p(X_3 \mid X_5)$	0	1
0	.4	.1
1	.6	.9

$$p(x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1, x_5 = 0) = 0.7 \cdot 0.6 \cdot 0.6 \cdot 0.2 \cdot 0.4 = 0.02016$$

Exhaustive inference: $p(x_2 = 0, x_5 = 1) = 0.258$

x_1	x_2	x_3	x_4	x_5	$p(\boldsymbol{x})$	x_1	x_2	x_3	x_4	x_5	$p(\boldsymbol{x})$
0	0	0	0	0	.01728	1	0	0	0	0	.05376
0	**0**	**0**	**0**	**1**	**.0003**	**1**	**0**	**0**	**0**	**1**	**.0126**
0	0	0	1	0	.00432	1	0	0	1	0	.01344
0	**0**	**0**	**1**	**1**	**.0003**	**1**	**0**	**0**	**1**	**1**	**.0126**
0	0	1	0	0	.02592	1	0	1	0	0	.08064
0	**0**	**1**	**0**	**1**	**.0027**	**1**	**0**	**1**	**0**	**1**	**.1134**
0	0	1	1	0	.00648	1	0	1	1	0	.02016
0	**0**	**1**	**1**	**1**	**.0027**	**1**	**0**	**1**	**1**	**1**	**.1134**
0	1	0	0	0	.06912	1	1	0	0	0	.03584
0	1	0	0	1	.0012	1	1	0	0	1	.0084
0	1	0	1	0	.01728	1	1	0	1	0	.00896
0	1	0	1	1	.0012	1	1	0	1	1	.0084
0	1	1	0	0	.10368	1	1	1	0	0	.05376
0	1	1	0	1	.0108	1	1	1	0	1	.0756
0	1	1	1	0	.02592	1	1	1	1	0	.01344
0	1	1	1	1	.0108	1	1	1	1	1	.0756

So, we need something smarter

Variable Elimination

Algorithm 3 `VE_PR1`($\mathcal{N}$, $\mathbf{Q}$, π)

input:

$\mathcal{N}$: Bayesian network
$\mathbf{Q}$: variables in network $\mathcal{N}$
π: ordering of network variables not in $\mathbf{Q}$

output: the prior marginal $\Pr(\mathbf{Q})$

main:

```
S ← CPTs of network N
for i = 1 to length of order π do
    f ← ∏_k f_k, where f_k belongs to S and mentions variable π(i)
    f_i ← ∑_π(i) f
    replace all factors f_k in S by factor f_i
end for
return ∏_{f∈S} f
```

Simple algorithm, but **efficiency depends on variable order** π!

Variable Elimination in Trees

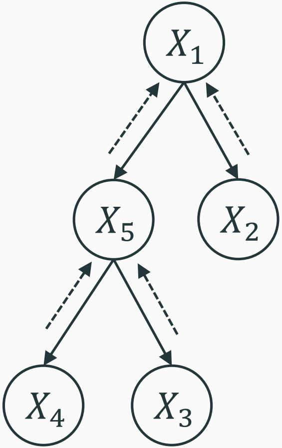
When using **reverse topological order** on a tree-structured BN:

- The order width[1] $w = 1$, so VE will run in $\mathcal{O}(n \exp(w)) = \mathcal{O}(n)$ time!
- Algorithm can be elegantly restructured as **message passing** method

[1] Recall: order width of π is largest number of variables in factor f_i on Line 4 of VE, for order π.

Message Passing

- Every non-root node sends messages to its parent
- Every node can send a message if and only if it has received messages from all its children
- We denote by $\mu_{X_i \to X_{\tau(i)};x}$ the message sent from X_i to its parent $X_{\tau(i)}$ when $X_{\tau(i)} = x$



Marginal Inference: The Sum-Product Algorithm

- Let $(\mathcal{T}, \mathcal{P})$ be a tree-shaped BN over $\boldsymbol{X} = \{X_i\}_{i=1}^{d}$ and X_r the root of $\mathcal{T}$
- We want to compute $p(\hat{\boldsymbol{x}})$ where $\hat{\boldsymbol{x}} \in \hat{\boldsymbol{\mathcal{X}}}$ and $\hat{\boldsymbol{X}} \subseteq \boldsymbol{X}$

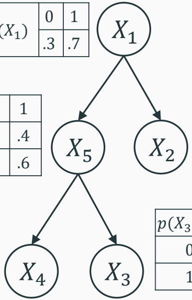
$$p(\hat{\boldsymbol{x}}) = \begin{cases} p(x_r = \hat{\boldsymbol{x}}_r) \prod_{X_j \in \mathrm{ch}(X_r)} \mu_{X_j \to X_r; \hat{\boldsymbol{x}}_r} & \text{if } X_r \in \hat{\boldsymbol{X}} \\ \sum_{x \in \mathcal{X}_r} p(x_r = x) \prod_{X_j \in \mathrm{ch}(X_r)} \mu_{X_j \to X_r; x} & \text{otherwise (VE)} \end{cases}$$

$$\mu_{X_i \to X_{\tau(i)}; x} = \begin{cases} p(x_i = \hat{\boldsymbol{x}}_i | x_{\tau(i)} = x) \prod_{X_j \in \mathrm{ch}(X_i)} \mu_{X_j \to X_i; \hat{\boldsymbol{x}}_i} & \text{if } X_i \in \hat{\boldsymbol{X}} \\ \sum_{x' \in \mathcal{X}_i} p(x_i = x' | x_{\tau(i)} = x) \prod_{X_j \in \mathrm{ch}(X_i)} \mu_{X_j \to X_i; x'} & \text{otherwise (VE)} \end{cases}$$

Marginal Inference: How to compute $p(x_2 = 0, x_5 = 1)$?

- We use $X_3 \succ X_4 \succ X_2 \succ X_5 \succ X_1$ as reversed topological order

$p(X_1)$	0	1
	.3	.7



$p(X_5 \mid X_1)$	0	1
0	.9	.4
1	.1	.6

$p(X_2 \mid X_1)$	0	1
0	.2	.6
1	.8	.4

$p(X_4 \mid X_5)$	0	1
0	.8	.5
1	.2	.5

$p(X_3 \mid X_5)$	0	1
0	.4	.1
1	.6	.9

$$\mu_{X_3 \to X_5;1} = 0.1 + 0.9 = \mathbf{1.0}$$

$$\mu_{X_4 \to X_5;1} = 0.5 + 0.5 = \mathbf{1.0}$$

$$\mu_{X_2 \to X_1;0} = 0.2 \qquad \mu_{X_2 \to X_1;1} = 0.6$$

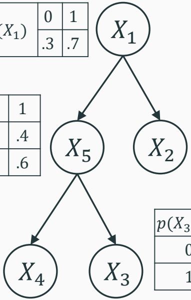
$$\mu_{X_5 \to X_1;0} = 0.1 \cdot \mathbf{1.0} \cdot \mathbf{1.0} = 0.1 \qquad \mu_{X_5 \to X_1;1} = 0.6 \cdot \mathbf{1.0} \cdot \mathbf{1.0} = 0.6$$

$$p(x_2 = 0, x_5 = 1) = p(x_1 = 0) \cdot \mu_{X_2 \to X_1;0} \cdot \mu_{X_5 \to X_1;0} + p(x_1 = 1) \cdot \mu_{X_2 \to X_1;1} \cdot \mu_{X_5 \to X_1;1}$$

$$= 0.3 \cdot (0.2 \cdot 0.1) + 0.7 \cdot (0.6 \cdot 0.6) = 0.258$$

Marginal Inference: How to compute $p(x_2 = 0, x_3 = 1, x_4 = 1)$?

- We use $X_3 \succ X_4 \succ X_2 \succ X_5 \succ X_1$ as reversed topological order



$p(X_1)$	0	1
	.3	.7

$p(X_5 \mid X_1)$	0	1
0	.9	.4
1	.1	.6

$p(X_2 \mid X_1)$	0	1
0	.2	.6
1	.8	.4

$p(X_4 \mid X_5)$	0	1
0	.8	.5
1	.2	.5

$p(X_3 \mid X_5)$	0	1
0	.4	.1
1	.6	.9

$$\mu_{X_3 \to X_5;0} = \mathbf{0.6} \qquad \mu_{X_3 \to X_5;1} = \mathbf{0.9}$$

$$\mu_{X_4 \to X_5;0} = \mathbf{0.2} \qquad \mu_{X_4 \to X_5;1} = \mathbf{0.5}$$

$$\mu_{X_2 \to X_1;0} = 0.2 \qquad \mu_{X_2 \to X_1;1} = 0.6$$

$$\mu_{X_5 \to X_1;0} = 0.9 \cdot \mathbf{0.6} \cdot \mathbf{0.2} + 0.1 \cdot \mathbf{0.9} \cdot \mathbf{0.5} = 0.153$$

$$\mu_{X_5 \to X_1;1} = 0.4 \cdot \mathbf{0.6} \cdot \mathbf{0.2} + 0.6 \cdot \mathbf{0.9} \cdot \mathbf{0.5} = 0.318$$

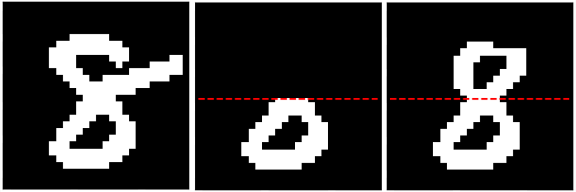
$$p(x_2 = 0, x_3 = 1, x_4 = 1) = p(x_1 = 0) \cdot \mu_{X_2 \to X_1;0} \cdot \mu_{X_5 \to X_1;0} + p(x_1 = 1) \cdot \mu_{X_2 \to X_1;1} \cdot \mu_{X_5 \to X_1;1}$$

$$= 0.3 \cdot (0.2 \cdot 0.153) + 0.7 \cdot (0.6 \cdot 0.318) = 0.14274$$

MPE Inference

- The Most Probable Explanation (MPE) task computes the most probable state of variables that do not have evidence
- The difference between standard inference and MPE inference is that instead of summing values, the **maximum** is used

Application: data imputation, e.g. **inpainting**



Exhaustive inference: What is the most probable state?

x_1	x_2	x_3	x_4	x_5	$p(\boldsymbol{x})$	x_1	x_2	x_3	x_4	x_5	$p(\boldsymbol{x})$
0	0	0	0	0	.01728	1	0	0	0	0	.05376
0	0	0	0	1	.0003	1	0	0	0	1	.0126
0	0	0	1	0	.00432	1	0	0	1	0	.01344
0	0	0	1	1	.0003	1	0	0	1	1	.0126
0	0	1	0	0	.02592	1	0	1	0	0	.08064
0	0	1	0	1	.0027	**1**	**0**	**1**	**0**	**1**	**.1134**
0	0	1	1	0	.00648	1	0	1	1	0	.02016
0	0	1	1	1	.0027	**1**	**0**	**1**	**1**	**1**	**.1134**
0	1	0	0	0	.06912	1	1	0	0	0	.03584
0	1	0	0	1	.0012	1	1	0	0	1	.0084
0	1	0	1	0	.01728	1	1	0	1	0	.00896
0	1	0	1	1	.0012	1	1	0	1	1	.0084
0	1	1	0	0	.10368	1	1	1	0	0	.05376
0	1	1	0	1	.0108	1	1	1	0	1	.0756
0	1	1	1	0	.02592	1	1	1	1	0	.01344
0	1	1	1	1	.0108	1	1	1	1	1	.0756

Exhaustive inference: What is the most probable state when $x_2 = 1$ and $x_5 = 0$?

x_1	x_2	x_3	x_4	x_5	$p(\boldsymbol{x})$	x_1	x_2	x_3	x_4	x_5	$p(\boldsymbol{x})$
0	0	0	0	0	.01728	1	0	0	0	0	.05376
0	0	0	0	1	.0003	1	0	0	0	1	.0126
0	0	0	1	0	.00432	1	0	0	1	0	.01344
0	0	0	1	1	.0003	1	0	0	1	1	.0126
0	0	1	0	0	.02592	1	0	1	0	0	.08064
0	0	1	0	1	.0027	1	0	1	0	1	.1134
0	0	1	1	0	.00648	1	0	1	1	0	.02016
0	0	1	1	1	.0027	1	0	1	1	1	.1134
0	**1**	**0**	**0**	**0**	**.06912**	**1**	**1**	**0**	**0**	**0**	**.03584**
0	1	0	0	1	.0012	1	1	0	0	1	.0084
0	**1**	**0**	**1**	**0**	**.01728**	**1**	**1**	**0**	**1**	**0**	**.00896**
0	1	0	1	1	.0012	1	1	0	1	1	.0084
0	**1**	**1**	**0**	**0**	**.10368**	**1**	**1**	**1**	**0**	**0**	**.05376**
0	1	1	0	1	.0108	1	1	1	0	1	.0756
0	**1**	**1**	**1**	**0**	**.02592**	**1**	**1**	**1**	**1**	**0**	**.01344**
0	1	1	1	1	.0108	1	1	1	1	1	.0756

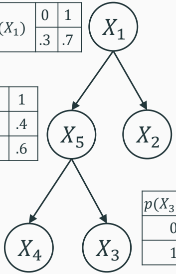
MPE Inference: The Max-Product Algorithm

- Let $(\mathcal{T}, \mathcal{P})$ be a tree-shaped BN over $\boldsymbol{X} = \{X_i\}_{i=1}^{d}$ and X_r the root of $\mathcal{T}$
- $\hat{\boldsymbol{x}} \in \hat{\boldsymbol{\mathcal{X}}}$, $\hat{\boldsymbol{X}} \subseteq \boldsymbol{X}$ and $\boldsymbol{Z} = \boldsymbol{X} \setminus \hat{\boldsymbol{X}}$
- We want to compute $\max_{\boldsymbol{z} \in \boldsymbol{\mathcal{Z}}} p(\hat{\boldsymbol{x}}, \boldsymbol{z}) \propto \max_{\boldsymbol{z} \in \boldsymbol{\mathcal{Z}}} p(\boldsymbol{z} | \hat{\boldsymbol{x}})$

$$\max_{\boldsymbol{z} \in \boldsymbol{\mathcal{Z}}} p(\hat{\boldsymbol{x}}, \boldsymbol{z}) = \begin{cases} p(x_r = \hat{\boldsymbol{x}}_r) \prod_{X_j \in \mathrm{ch}(X_r)} \tilde{\mu}_{X_j \to X_r; \hat{\boldsymbol{x}}_r} & \text{if } X_r \in \hat{\boldsymbol{X}} \\ \max_{x \in \mathcal{X}_r} p(x_r = x) \prod_{X_j \in \mathrm{ch}(X_r)} \tilde{\mu}_{X_j \to X_r; x} & \text{otherwise} \end{cases}$$

$$\tilde{\mu}_{X_i \to X_{\tau(i)}; x} = \begin{cases} p(x_i = \hat{\boldsymbol{x}}_i | x_{\tau(i)} = x) \prod_{X_j \in \mathrm{ch}(X_i)} \tilde{\mu}_{X_j \to X_i; \hat{\boldsymbol{x}}_i} & \text{if } X_i \in \hat{\boldsymbol{X}} \\ \max_{x' \in \mathcal{X}_i} p(x_i = x' | x_{\tau(i)} = x) \prod_{X_j \in \mathrm{ch}(X_i)} \tilde{\mu}_{X_j \to X_i; x'} & \text{otherwise} \end{cases}$$

MPE Inference: What is the most probable state?



$p(X_1)$	0	1
	.3	.7

$p(X_5 \mid X_1)$	0	1
0	.9	.4
1	.1	.6

$p(X_2 \mid X_1)$	0	1
0	.2	.6
1	.8	.4

$p(X_4 \mid X_5)$	0	1
0	.8	.5
1	.2	.5

$p(X_3 \mid X_5)$	0	1
0	.4	.1
1	.6	.9

$$\tilde{\mu}_{X_3 \to X_5;0} = \max[.4, .6] = .6 \quad [\![1]\!] \qquad \tilde{\mu}_{X_3 \to X_5;1} = \max[.1, .9] = .9 \quad [\![1]\!]$$

$$\tilde{\mu}_{X_4 \to X_5;0} = \max[.8, .2] = .8 \quad [\![0]\!] \qquad \tilde{\mu}_{X_4 \to X_5;1} = \max[.5, .5] = .5 \quad [\![0]\!]$$

$$\tilde{\mu}_{X_2 \to X_1;0} = \max[.2, .8] = .8 \quad [\![1]\!] \qquad \tilde{\mu}_{X_2 \to X_1;1} = \max[.6, .4] = .6 \quad [\![0]\!]$$

$$\tilde{\mu}_{X_5 \to X_1;0} = \max[(.9 \cdot .6 \cdot .8), (.1 \cdot .9 \cdot .5)] = .432 \quad [\![0]\!]$$

$$\tilde{\mu}_{X_5 \to X_1;1} = \max[(.4 \cdot .6 \cdot .8), (.6 \cdot .9 \cdot .5)] = .27 \quad [\![1]\!]$$

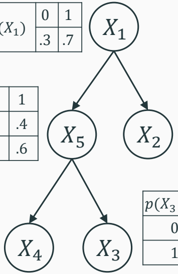
$$\max_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} p(\boldsymbol{x}) = \max[(p(x_1 = 0) \cdot \mu_{X_2 \to X_1;0} \cdot \mu_{X_5 \to X_1;0}), (p(x_1 = 1) \cdot \mu_{X_2 \to X_1;1} \cdot \mu_{X_5 \to X_1;1})]$$

$$= \max[(0.3 \cdot 0.8 \cdot 0.432), (0.7 \cdot 0.6 \cdot 0.27)] = 0.1134 \quad [\![1]\!]$$

$$x_1 = 1 \implies x_2 = 0 \text{ and } x_5 = 1 \qquad x_5 = 1 \implies x_3 = 1 \text{ and } x_4 = 0$$

MPE Inference: What is the most probable state when $x_2 = 1$ and $x_5 = 0$?

$p(X_1)$	0	1
	.3	.7



$p(X_5 \mid X_1)$	0	1
0	.9	.4
1	.1	.6

$p(X_2 \mid X_1)$	0	1
0	.2	.6
1	.8	.4

$p(X_4 \mid X_5)$	0	1
0	.8	.5
1	.2	.5

$p(X_3 \mid X_5)$	0	1
0	.4	.1
1	.6	.9

$$\tilde{\mu}_{X_3 \to X_5;0} = \max[.4, .6] = \mathbf{.6} \quad \llbracket 1 \rrbracket$$

$$\tilde{\mu}_{X_4 \to X_5;0} = \max[.8, .2] = \mathbf{.8} \quad \llbracket 0 \rrbracket$$

$$\tilde{\mu}_{X_2 \to X_1;0} = .8 \quad \llbracket 1 \rrbracket \qquad \tilde{\mu}_{X_2 \to X_1;1} = .4 \quad \llbracket 1 \rrbracket$$

$$\tilde{\mu}_{X_5 \to X_1;0} = (.9 \cdot \mathbf{.6} \cdot \mathbf{.8}) = .432 \quad \llbracket 0 \rrbracket$$

$$\tilde{\mu}_{X_5 \to X_1;1} = (.4 \cdot \mathbf{.6} \cdot \mathbf{.8}) = .192 \quad \llbracket 0 \rrbracket$$

$$\max_{\mathbf{z} \in \mathcal{Z}} p(\hat{\mathbf{x}}, \mathbf{z}) = \max[(p(x_1 = 0) \cdot \mu_{X_2 \to X_1;0} \cdot \mu_{X_5 \to X_1;0}), (p(x_1 = 1) \cdot \mu_{X_2 \to X_1;1} \cdot \mu_{X_5 \to X_1;1})]$$

$$= \max[(0.3 \cdot 0.8 \cdot 0.432), (0.7 \cdot 0.4 \cdot 0.192)] = 0.10368 \quad \llbracket 0 \rrbracket$$

$$x_1 = 0 \text{ and } x_2 = 1 \text{ and } x_5 = 0 \qquad x_5 = 0 \implies x_3 = 1 \text{ and } x_4 = 0$$

Efficient inference with VE using **reverse topological order**

- This has order width $w = 1$, so VE then has complexity $\mathcal{O}(n)$

Algorithm can be restructured as **message passing** method

- Sum-Product algorithm for marginal inference
- Max-Product algorithm for MPE

Ancestral Sampling

Ancestral Sampling

Method to draw i.i.d. samples $\mathbf{x} \sim p(\mathbf{X})$, where $p(\mathbf{X})$ is (encoded by) a BN $(\mathcal{G}, \mathcal{P})$

- Requires method to sample $x \sim p(X \mid \mathbf{pa}(X))$ for each X
  - E.g. inverse-transform sampling

Ancestral Sampling

Method to draw i.i.d. samples $\mathbf{x} \sim p(\mathbf{X})$, where $p(\mathbf{X})$ is (encoded by) a BN $(\mathcal{G}, \mathcal{P})$

- Requires method to sample $x \sim p(X \,|\, \mathbf{pa}(X))$ for each X
  - E.g. inverse-transform sampling

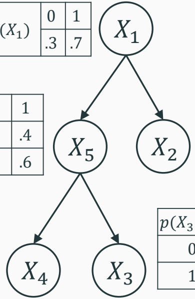
Simply use **topological order** π of $\mathcal{G}$. For each $i = 1, \ldots, |\pi|$,

- Let $\rho_i = \mathbf{pa}(X_{\pi(i)})$
- Sample $x_{\pi(i)} \sim p(X_{\pi(i)} \,|\, \mathbf{X}_{\rho_i} = \mathbf{x}_{\rho_i})$, where $\mathbf{x}_{\rho_i}$ are values already sampled

Then $\mathbf{x} \sim p(\mathbf{X})$

Ancestral Sampling

- Let $X \sim \mathcal{B}(p)$ a Bernoulli RV with probability p. To sample from X we generate a random number $\epsilon \in [0,1]$ if $\epsilon \leq p$ then $x = 1$ else $x = 0$.
- We use $X_1 \prec X_2 \prec X_5 \prec X_3 \prec X_4$ as topological order.



$p(X_1)$	0	1
	.3	.7

$p(X_5 \mid X_1)$	0	1
0	.9	.4
1	.1	.6

$p(X_2 \mid X_1)$	0	1
0	.2	.6
1	.8	.4

$p(X_4 \mid X_5)$	0	1
0	.8	.5
1	.2	.5

$p(X_3 \mid X_5)$	0	1
0	.4	.1
1	.6	.9

1. `rand([0, 1])` $= 0.8 \rightarrow x_1 = 0$
2. `rand([0, 1])` $= 0.3 \rightarrow x_2 = 1$
3. `rand([0, 1])` $= 0.5 \rightarrow x_5 = 0$
4. `rand([0, 1])` $= 0.1 \rightarrow x_3 = 1$
5. `rand([0, 1])` $= 0.6 \rightarrow x_4 = 0$

Summary and Outlook

Today's lecture

- Chow-Liu Trees
- Inference in tree-shaped BNs
- Ancestral sampling

Next lecture

- Markov networks
- missing data