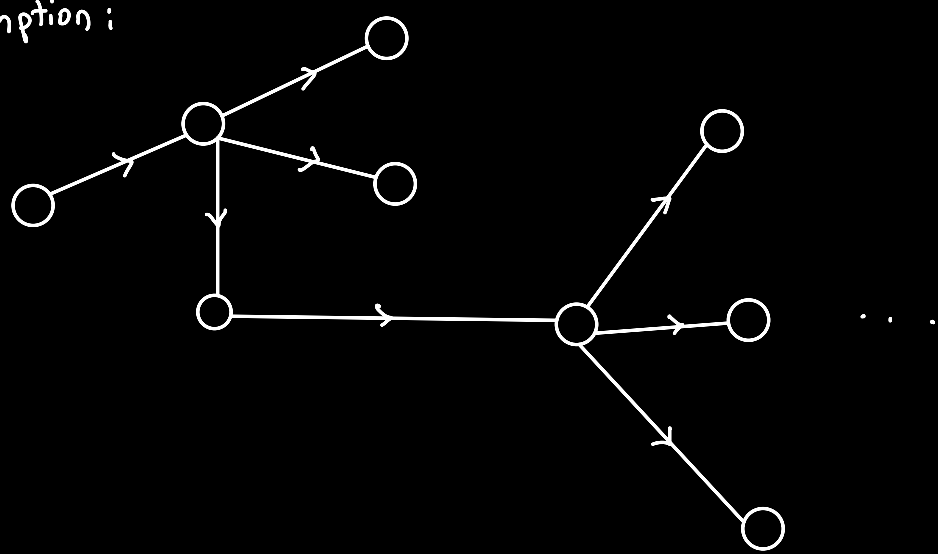
Learning Bayesian Networks defined by trees.

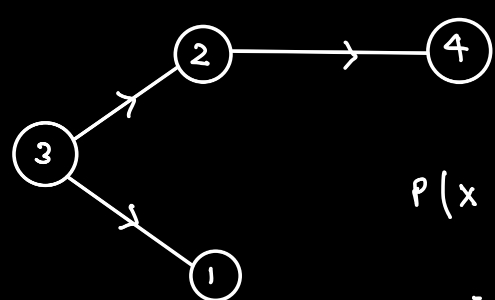
Distribution D : $(x_1, x_2, \ldots x_d) \in \Sigma^n$ $(\{0,1\}^d)$

Assumption:



Rooted tree: Each node (except root) has exactly one parent.

Example:



$$P(X = x_1, x_2, x_3, x_4)$$

$$= P(X_3 = x_3) \cdot Pr(X_2 = x_2 \mid X_3 = x_3)$$

$$\cdot Pr(X_1 = x_1 \mid X_3 = x_3)$$

$$\cdot Pr(X_4 = x_4 \mid X_2 = x_2).$$

Suppose we see samples from a Bayes net as above, can you learn the distribution.

Structure Learning

* "learn the underlying graph"

learn the tree

Parameter Learning

learn the conditional distributions.

Goal:

Let us try to learn a distribution D' such that $\text{dist}(D, D')$ is small).

KL - DIVERGENCE:

"Measures distance between distributions"

We have two distributions p, q over some space Ω.

(cross entropy)

$$KL(p \| q) = \sum_{s \in \Omega} p(s) \log \frac{p(s)}{q(s)}$$

Probability s happens under q.

Probability s happens under p

EXAMPLE:

$$\Omega = \{0,1\}$$

$$p(0) = 1/2, \quad p(1) = 1/2$$

$$q(0) = 3/4, \quad q(1) = 1/4$$

$$KL(p|q) = p(0) \cdot \log \frac{p(0)}{q(0)} + p(1) \cdot \log \frac{p(1)}{q(1)}$$

$$= \frac{1}{2} \cdot \log\left(\frac{2}{3}\right) + \frac{1}{2} \cdot \log(2)$$

$$= \frac{1}{2} \cdot \log\left(\frac{4}{3}\right)$$

What is $KL(p|p)$?

$\rightarrow 0$

Property:

$KL(p|q) \geq 0$ and is $0 \iff p = q$.

NOT SYMMETRIC!

INPUT: Samples from a distribution P on Σ^d ($\Sigma = \{0,1\}$)

(being generated by some unknown Bayes net which is a tree: T^*)

OUTPUT: Find a tree T and a corresponding Bayes net P_T such that $KL(P|P_T) \leq \varepsilon$.

EXAMPLE:

$$5 \to 3 \to 2 \to 4 \to 1 \qquad T^*$$

$$(x_1, x_2, x_3, x_4, x_5)$$

If I give a tree T and ask find me the Bayes net P_T that minimizes $KL(P|P_T)$.

$$\begin{tikzcd}
1 \arrow[r] & 5 \arrow[r] \arrow[dr] & 3 \arrow[r] & 2 \\
 & & 4 &
\end{tikzcd}$$

Idea: Well, use the "empirical" conditional probabilities.

1. Estimate the conditional probabilities along the tree T.

(e.g: Estimate $Pr[x_1 = 1] \cdot Pr[x_5 = 1 \mid x_1 = 0]$, $Pr[x_5 = 1, x_1 = 1]$)

$\vdots$

INPUT: Samples from P and a tree T.

Algorithm: For each edge (i,j) in T, estimate $Pr[x_j \mid x_i]$ and use these to define P_T.

CHOW - LIU ALGORITHM:

→ For any tree T,

$$KL(P \| P_T) = J_P - \sum_{(i,j) \text{ is an edge in } T} I(x_i ; x_j)$$

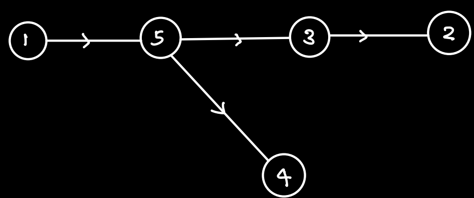
J_P ↓ Some number depending on P

$$5 \to 3 \to 2 \to 4 \to 1 \qquad T^*$$

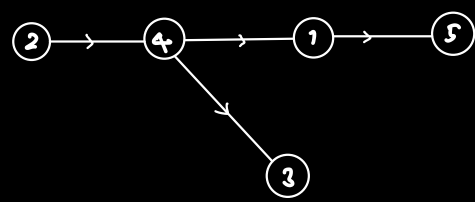
$(x_1, x_2, x_3, x_4, x_5)$

Possible trees:

$D(P|P_T)$



$D(P|P_T)$



...

Given a tree T, we can find P_T.

$I(x_i; x_j) \equiv$ "Measures how much information about x_j does x_i have".

$$I(x_i; x_j) = \sum_{a,b} Pr[x_i = a, x_j = b] \cdot \log \frac{Pr[x_i = a]\, Pr[x_j = b]}{Pr[x_i = a, x_j = b]}$$

We can estimate $I(x_i, x_j)$ from sample.

$$5 \to 3 \to 2 \to 4 \to 1 \qquad T^*$$

$$(x_1, x_2, x_3, x_4, x_5)$$

$$T: \begin{tikzcd} 1 \arrow[r] & 5 \arrow[r] \arrow[dr] & 3 \arrow[r] & 2 \\ & & 4 & \end{tikzcd}$$

Chow-Liu Bound (1968):

$$KL(P \| P_T) = J_P - I(x_1; x_5) - I(x_5; x_3) - I(x_5; x_4)$$
$$- I(x_3; x_2).$$

Summary: Find T that minimizes $KL(P \| P_T)$ or find

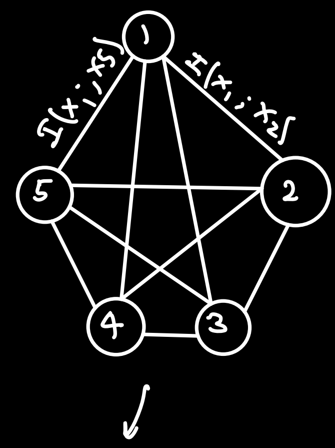
$$\arg\min_T \; J_P - \sum_{(i,j)\in T} I(x_i; x_j)$$

$$\equiv \arg\min_T \sum_{(i,j)\in T} - I(x_i; x_j)$$

$$\equiv \arg\max_T \sum_{(i,j)\in T} I(x_i; x_j)$$

IDEA:

We have samples from P $\rightsquigarrow$



Maximum Spanning Tree.

CHOW - LIU ALGORITHM:

- Use samples to estimate $I(x_i ; x_j)$ for all i, j.
- Form a weighted graph where weights are exactly $I(x_i ; x_j)$.
- Compute the maximum spanning tree T' in G.
- Output $P_{T'}$.

Maximum Spanning Tree: Given a weighted graph G on d vertices, find tree T that has maximum total weight. - We can do this very fast. We can do this in linear time.

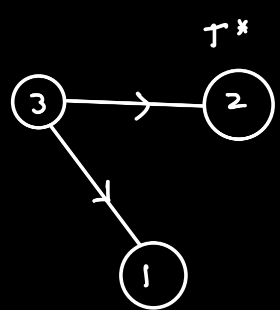
$(x_1, x_2, \dots, x_5)$: $\square \to$ Compute $I(x_1; x_2)$

$\square \to$ Compute $I(x_1; x_3)$

$\square \to$ Compute $I(x_1; x_4)$

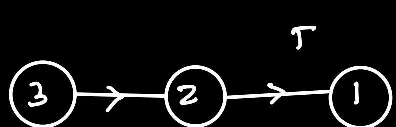
$\vdots$

Example:



True distribution, P

$$P[X = x_1, x_2, x_3] = P[X_3 = x_3] \cdot P[X_2 = x_2 \mid X_3 = x_3] \cdot P[X_1 = x_1 \mid X_3 = x_3]$$



$$P_T[X = x_1, x_2, x_3] = P[X_3 = x_3] \cdot P[X_2 = x_2 \mid X_3 = x_3] \cdot P[X_1 = x_1 \mid X_2 = x_2]$$

$\searrow$ P_T.

Summary: "DIRECTED GRAPHICAL MODELS"

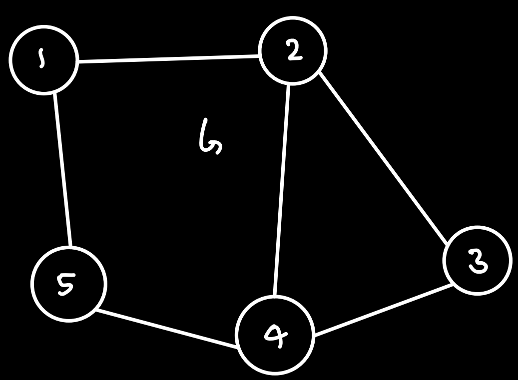
- Can learn tree-structured Bayes networks
- Active research: What are other structural assumptions
  Can we learn Bayes networks.

Undirected Graphical Models:

- "Markov Random Fields" (MRFs).

Distribution: $D \equiv (x_1, x_2, \dots x_d)$

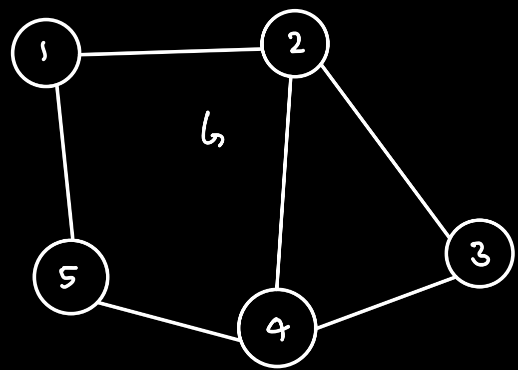
Dependency graph G for D.



D satisfies "Pairwise Markov Property" with respect to G if

If i, j have no edge, then $x_i \perp x_j \mid x_{rest}$

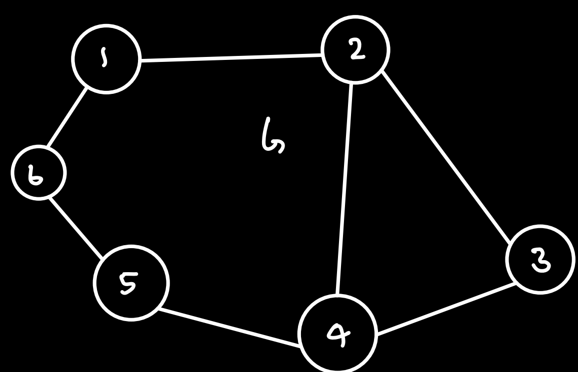
eg: $x_1 \perp x_4 \mid x_2, x_3, x_5$



D satisfies "Local Markov Property" with respect to G if

If i, j have no edge, then $x_i \perp x_j \mid x_{\{\text{neighbors of } i\}}$

eg: $x_1 \perp x_4 \mid x_2, x_5$



D satisfies "Global Markov Property" with respect to G if

If i, j have no edge, then $x_i \perp x_j \mid x_{\{\text{any separating set}\}}$

subset of vertices removing which disconnects i and j in G.

eg: $x_1 \perp x_4 \mid x_2, x_5$

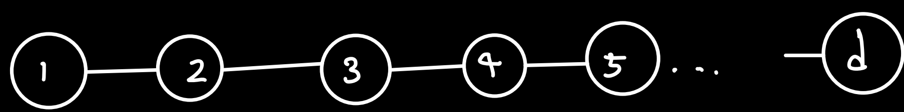
Global MP $\Rightarrow$ Local MP $\Rightarrow$ Pairwise MP

For "most reasonable" distributions all three are equivalent.

Example:

Markov Chain

$x_1, x_2, \ldots, x_d$ $\quad x_{i+1} \perp x_{i-1} \mid x_i$



Remark: We will say D has dependency graph G if it satisfies Markov property with respect to G.

MAIN LEARNING CHALLENGES:

Structure Learning: Given samples from D learn its dependency graph G.

Parametric Learning: Given samples from D learn the full distribution.

Inference: You know dependency graph want to find most likely value of $x_i \mid x_{\{\text{partial assignment}\}}$.

MAP

GOAL: Structure Learning

INPUT: Samples $x^1, x^2, \dots, x^n \sim D$

OUTPUT: Dependency graph of D.

ILL-POSED: Complete graph is a dependency graph for all distributions!

"Minimal Dependency graph?"

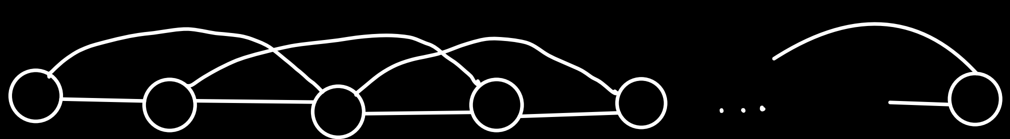
ASSUMPTION:

Dependency graph of D is sparse.

- Each vertex has a bounded degree k.

INPUT : Samples $x^1, x^2, \dots, x^n \sim D$

OUTPUT : A dependency graph for D where each vertex has degree $\le k$.



$k = 4$

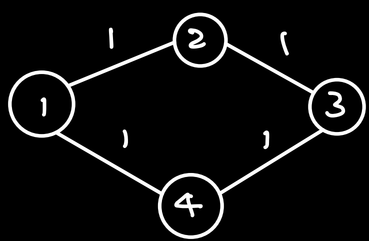
→ If we know the structure, the general learning problem becomes easier.

Learning Boltzmann Machines

$\downarrow$

D on $\{1,-1\}^d$

$$\Pr[x = x] \propto \exp\Big(\sum_{\{i,j\} \in G} w_{ij} x_i x_j\Big)$$



$$\Pr[x = x] \propto \exp(x_1 x_2 + x_2 x_3 + x_3 x_4 + x_4 x_1)$$

Distributions as defined above, they satisfy Markov property with respect to G.

Gaussian Graphical Models

$\downarrow$

D on $\mathbb{R}^d$

Distribution D is $N(0, \Sigma)$

Σ is the covariance matrix.

$$X \sim N(0, \Sigma)$$

$$\Sigma_{ij} = \mathbb{E}[x_i x_j]$$

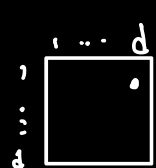
$\Sigma_{ij} = 0 \Rightarrow x_i, x_j$ are independent.

(Dempsey 1972):

Precision Matrix

$$\Theta = \Sigma^{-1}$$

THEOREM: Gaussian distribution has dependency graph with support (Θ).



$i, j \in G$ if $\Theta_{ij} \neq 0$.