* Overview

- Graphical Models & Markov Properties
- Inference Problems : given $P_G(x)$ find
  - $P(x_i)$
  - $\arg\max_x P(x)$
  - compute Z.

  - Belief Propagation
  - Variational methods
  - Gibbs sampling

- Learning Graphical models : given samples $x^{(1)}, x^{(2)}, \dots, x^{(N)} \in \mathcal{X}^n$
  - Learn the structure of the graph
  - Learn the parameters of the factors.

* How to find Z from a black-box that gives $P(x_i)$ from $P(x)$.

* Structural learning.

- $X \in \mathcal{X}^n$ Random Vector.
- directed graphical model : $|E| = \binom{n}{2} = \frac{n(n-1)}{2}$

  $\#$ of possible DAGs $\leq 3^{|E|} = 3^{\frac{n(n-1)}{2}}$

- Given $X^{(1)}, \cdots, X^{(N)}$ independent samples from unknown $P(X)$.
  - How do we score each graph?
  - How do we find the graph with highest score?

- There are 2 ways to approach such statistical problems

Frequentist

- Assume graph G and conditionals $P = \{P(X_i | X_{\pi_i})\}_{i=1}^{n}$ are deterministic but unknown
- Maximum Likelihood (ML) estimation finds (G, P) that maximizes log likelihood

$$\max_{G,P} \sum_{j=1}^{N} \log \mathbb{P}_{G,P}(X^{(j)})$$

Bayesian

- Assume graph G and conditionals P are drawn from some known prior distribution $P_{G,P}(G, P)$
- Maximum a posteriori (MAP) estimation finds (G,P) that maximizes the posterior distribution

$$\max_{G,P} \mathbb{P}(G, P | X^{(1)}, X^{(2)}, \cdots, X^{(N)})$$

* Frequentist's approach to structural learning

$$\hat{G} = \arg\max_{G} \max_{P} \frac{1}{N} \sum_{i=1}^{N} \log P_{G,P}(X^{(i)})$$

*Simple Case with $n=2$, $X=(X_1, X_2) \in \{0,1\}^2$

samples $(0,0), (0,1), (1,1), (0,0)$

empirical distribution: $\hat{P}_1(x_1) = \begin{cases} 3/4, & x_1=0 \\ 1/4, & x_1=1 \end{cases}$, $\hat{P}_2(x_2) = \begin{cases} 1/2 & x_2=0 \\ 1/2, & x_2=1 \end{cases}$

case 1: for $G_1 =$ ① ②

the maximum likelihood estimate of $P_1(x_1), P_2(x_2)$ are

$$\max_{P_1} \frac{1}{N}\sum_{j=1}^{N} \log P_1(x_1^{(j)})$$

$$= \max_{P_1} \hat{P}_1(0)\cdot \log P_1(0) + \hat{P}_1(1)\cdot \log P_1(1)$$

$$= \max_{P_1} \underbrace{\sum_{x_1} \hat{P}_1(x_1) \log \hat{P}(x_1)}_{-H(\hat{P}_1)} + \underbrace{\sum_{x_1} \hat{P}_1(x_1) \log \frac{P_1(x_1)}{\hat{P}_1(x_1)}}_{-D_{KL}(\hat{P}_1 \| P_1)}$$

is maximized when $P_1 = \hat{P}_1$

*empirical distribution is the maximum likelihood.

for G_1,

$$\max_{P_1 \times P_2} \frac{1}{N}\sum_{j=1}^{N} \log P_1(x_1^{(j)}) P_2(x_2^{(j)})$$

$$= \underbrace{\sum_{x_1} \hat{P}_1(x_1)\cdot \log \hat{P}_1(x_1)}_{-H(\hat{P}_1)} + \underbrace{\sum_{x_2} \hat{P}_2(x_2) \log \hat{P}_2(x_2)}_{-H(\hat{P}_2)}$$

Case 2: $G_2 =$ ① $\rightarrow$ ②

$$\max_{P_{12}(x_1,x_2)} \frac{1}{N}\sum_{j=1}^{N} \log P(x^{(j)}) = -H(\hat{P}_{12}) \underbrace{- D_{KL}(\hat{P}_{12} \| P_{12})}_{\text{maximum achieved with } \hat{P}_{12} = P_{12}}$$

$$= -H(\hat{P}_{12})$$

$$\mathcal{L}(1 \quad 2) = -H(\hat{P}_1) - H(\hat{P}_2)$$

$$\mathcal{L}(1 \to 2) = -H(\hat{P}_{12})$$

· Remark 1: $-H(\hat{P}_{12}) \geqq -H(\hat{P}_1) - H(\hat{P}_2)$

hence, blindly choosing a more likely model results in overfitting.

↑ even if the true distribution was independent, we always choose dependent model.

· Remark 2: depending on the sample size N and the target false positive rate β, decision is made by

$$\mathcal{L}(1 \to 2) - \mathcal{L}(1 \quad 2) = H(\hat{P}_{12}) - H(\hat{P}_1) - H(\hat{P}_2)$$

$$\triangleq I_{\hat{P}_{12}}(X_1; X_2)$$

output $1 \to 2$ if $I_{\hat{P}_{12}}(X_1; X_2) > \frac{t_\beta}{n}$

$1 \quad 2$ otherwise.

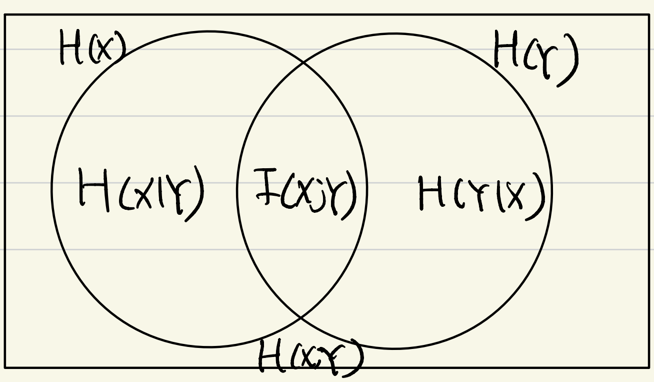
* We need to restrict the model class OR control false discovery rate

Refresh notations:

$$I(X;Y) \triangleq \sum_{X,Y} P(X,Y) \log \frac{P(X,Y)}{P(X) \cdot P(Y)}$$

$$H(X) \triangleq \sum_{X} -P(X) \log P(X)$$

$$H(Y|X) \triangleq \sum_{X,Y} -P(X,Y) \log P(Y|X)$$



$$H(Y|X) = H(Y) - I(X;Y)$$

$$H(X) + H(Y) = H(X,Y) + I(X;Y)$$

* Maximum Likelihood Approach for a DAG.

$$G^* = \arg\max_G \underbrace{\max_{\{P_i(X_i|X_{\pi_i})\}} \frac{1}{N}\sum_{j=1}^{N} \log \prod_{i=1}^{n} P_i(X_i|X_{\pi_i})}_{\text{the maximum is achieved at}}$$

$$P_i(X_i|X_{\pi_i}) = \hat{P}_i(X_i|X_{\pi_i})$$

the empirical distribution

$$\frac{1}{N}\sum_{j=1}^{N} \log \prod_{i=1}^{n} \hat{P}_i(X_i|X_{\pi_i})$$

$$= \sum_{i=1}^{n} \frac{1}{N}\sum_{j=1}^{N} \log \hat{P}_i(X_i|X_{\pi_i})$$

$$= \sum_{i=1}^{n} \sum_{X_i, X_{\pi_i}} \hat{P}_i(X_i, X_{\pi_i}) \cdot \log \hat{P}_i(X_i|X_{\pi_i})$$

$$= \sum_{i=1}^{n} -H_{\hat{p}}(X_i | X_{\pi_i})$$

$$= \sum_{i=1}^{n} \left\{ I_{\hat{p}}(X_i ; X_{\pi_i}) - \underbrace{H_{\hat{p}}(X_i)}_{\text{Does not depend on } G.} \right\}$$

find G from a family of graphs that maximize this term

*Remark: - this gives a "score" = likelihood for any given DAG G.

- we can now search over a class of graphs to find the best one.
- $I_{\hat{p}}(X_i ; X_{\pi_i}) \geq I_{\hat{p}}(X_i ; X_{\pi_i'})$ if $\pi_i \supset \pi_i'$ and hence denser graphs are preferred (and overfitted)
- So we need an appropriate class of graphs to search over.

* Chow-Liu algorithm: searches over all trees, efficiently.

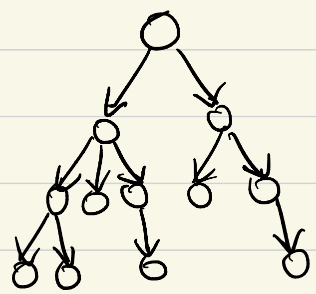
Step 1: Create a complete graph over $V = \{1, \cdots, n\}$ with the edge weights $I_{\hat{p}}(X_i, X_j) = w_{ij}$

Step 2: Use Kruskal's algorithm, for example, to find the max-weight spanning tree.

Claim: Chow-Liu algorithm finds the optimal tree that maximizes

$$\max_{G \in \text{Tree}} \sum_{i=1}^{n} I_{\hat{p}}(X_i ; X_{\pi_i})$$

proof: each node only has one parent.



for general graphs, $n!$ orderings make it intractable.

*Another impractical approach for Undirected graph learning.

Consider learning an Ising Model (G,θ), $\mathcal{X}=\{\pm 1\}$ from samples $\{x^{(1)}, x^{(2)}, \dots, x^{(N)}\} = D$

the likelihood is

$$P_{(G,\theta)}(D) = \prod_{l=1}^{N} P_{(G,\theta)}(x^{(l)})$$

$$= \prod_{l=1}^{N} \frac{1}{Z_G(\theta)} \prod_{(i,j)\in E} e^{x_i^{(l)} x_j^{(l)} \theta_{ij}} \prod_{i\in V} \cdot e^{x_i^{(l)} \cdot \theta_i}$$

$$= \exp\Big\{-N \cdot \log Z_G(\theta) + \sum_{(i,j)\in E} N \cdot \underbrace{\hat{M}_{ij}}_{= \frac{1}{N}\sum_{l=1}^{N} x_i^{(l)} x_j^{(l)}} + \sum_{i\in V} N \cdot \underbrace{\hat{M}_{ii}}_{= \frac{1}{N}\sum_{l=1}^{N} x_i^{(l)}} \Big\}$$

the log-likelihood is

$$L(G,\theta,D) = -\frac{1}{N} \log P_{(G,\theta)}(D)$$

$$= \underbrace{\Phi(\theta)}_{\text{log-partition function}} - \langle \hat{M}, \theta \rangle$$

$$\hat{M} = \begin{bmatrix} \hat{M}_{11} & \hat{M}_{12} & \cdots \\ & & \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_1 & \theta_{12} & \cdots \\ \vdots & & \\ & & \end{bmatrix}$$

non-zero for $(i,j)\in E$ & (i,i).

Remark: this is strictly convex in θ,
but log-partition function requires inference.

* learning is easier when inference is easier.
but in general this is computationally intractable, even if the graph is sparse, requiring $O(|\mathcal{X}|^n)$ computations.

but continuing our (theoretical) investigation,

we want to apply this method to learn the structure of the graph as follows

$$\underset{G}{\text{minimize}}\ \underset{\theta}{\text{minimize}}\ \mathcal{L}(G, \theta, D)$$

As we did previously, we need to restrict our search to a class of "simple" graphs, as otherwise dense graphs always win. A natural condition is $|E| \leq m$.

$$\underset{\theta}{\text{minimize}}\ \mathcal{L}(K_n, \theta, D) \qquad \text{s.t.} \quad \|\theta\|_0 \leq m$$

, where K_n is the complete graph

As $\|\theta\|_0$ constraint is intractable, people have proposed

$$\underset{\theta}{\text{minimize}}\ \Phi(\theta) - \langle \hat{\mu}, \theta \rangle + \lambda \cdot \|\theta\|_1.$$

where $\|\theta\|_1 = \sum_{i,j} |\theta_{ij}|$

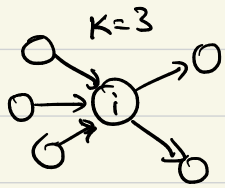
A different approach: Local Independence Test for undirected graphical models.

tries to take advantage of sparsity of the learned graph. to make learning faster than $O(|\mathcal{X}|^n)$.

Alg 1: Local Independence Test (samples $\{X^{(l)}\}_{l=1}^{N}$, neighborhood size K)

- $E = \phi$
- For each $i \in V$
- For each $S \subseteq V \setminus \{i\}$ s.t. $|S| \leq K$
- compute $SCORE(S, i) = H_{\hat{p}}(X_i | X_S)$
- Set $S^* \longleftarrow \arg\min_S SCORE(S, i)$
- $E \longleftarrow E \cup \{(i,j)\}_{j \in S^*}$

- Prune the resulting graph.



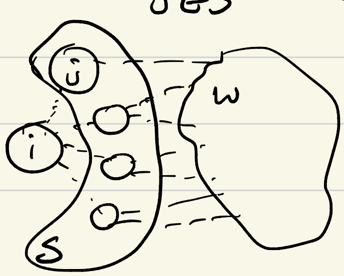
Remark 1: $X_i \perp\!\!\!\perp X_{rest} | X_S \iff$

$$H(X_i | X_S \cup X_{rest}) = H(X_i | X_S) \leq H(X_i | X_T) \quad \forall T \subset S.$$

Remark 2: Other scores have been proposed.

$$S^* = \arg\max \{ |S| : \varepsilon < SCORE(S, i) \}$$

$$SCORE(S,i) = \min_{\substack{W \subseteq V \setminus S \\ j \in S}} \max_{\substack{x_i, x_W \\ x_S, x_j = a}} \left| \hat{P}\{X_i = x_i | X_W = x_W, X_S = x_S\} - \hat{P}\{X_i = x_i | X_W = x_W, X_{S \setminus j} = x_{S \setminus j}, X_j = a\} \right|$$



Remark: $SCORE(S, i)$ will be small if it includes any non-significant node.

These approaches are still of more theoretical interest, as the run-time is $O(n^{k+1})$.

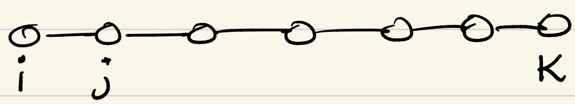
Here is a practical algorithm.

Alg 2: Thresholding (samples $\{X^{(l)}\}_{l=1}^{N}$, threshold τ)

- Compute the empirical correlation $\{\hat{M}_{ij}\}_{(i,j)\in V\times V}$
- For each $(i,j)\in V\times V$
- If $\hat{M}_{ij}\geq \tau$, set $(i,j)\in E$

where $\hat{M}_{ij} = \frac{1}{N}\sum_{l=1}^{N}\left(X_i^{(l)}-\bar{X}_i\right)\left(X_j^{(l)}-\bar{X}_j\right)$, $\bar{X}_i=\frac{1}{N}\sum_{l=1}^{N}X_i^{(l)}$

Remark: a heuristic based on the fact that two nodes faraway in the graph might be less correlated.



$\hat{M}_{iK}\ll\hat{M}_{ij}$

in general, this can fail if.

True graph learned graph