

# **Bayesian Networks IV: BN Structure Learning**

**For Self-learning Purposes**

**Jiashu Chen**

March 14, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Minimal I-map and P-map . . . . .	1
1.2	Covid Example . . . . .	1
1.2.1	Method 1: Check Independencies . . . . .	2
1.2.2	Method 2: Maximizing the Likelihood . . . . .	2
1.3	Structure Learning Approaches . . . . .	4
1.4	Summary: What is the Goal? . . . . .	4
<b>2</b>	<b>Constraint-based Algorithm</b>	<b>5</b>
2.1	From Distribution to Graph (Finding P-map) . . . . .	5
2.2	Finding the P-map Skeleton . . . . .	5
2.3	The Covid Problem . . . . .	6
2.4	Finding PDAG . . . . .	10
2.4.1	Algorithm: Discovering the class PDAG of a P-map of a Distribution . . . . .	10
2.4.2	Order of Complexity . . . . .	11
2.5	Summary: What is the Goal . . . . .	11
<b>3</b>	<b>Finding Independencies from Data</b>	<b>12</b>
3.1	Independence Test: Null Hypothesis Test . . . . .	12
3.2	Chi-squared Distribution . . . . .	12
3.3	Independence Test for 3 Variables . . . . .	14
3.4	Summary: What is the Goal . . . . .	14
<b>4</b>	<b>Score-based Algorithm</b>	<b>15</b>
4.1	The Likelihood Score . . . . .	15
4.1.1	Recall: Entropy and Mutual Information . . . . .	15
4.1.2	Likelihood Score Proposition . . . . .	16
4.1.3	The Covid Example . . . . .	16
4.1.4	Interpretation: Maximizing Dependence on Parents . . . . .	17
4.2	The Bayesian Score . . . . .	17
4.3	The BIC Score . . . . .	18

4.3.1	The Covid Example . . . . .	18
4.4	The Learning Procedure . . . . .	19
4.4.1	Score-based Algorithm . . . . .	19
4.4.2	Search Procedure: Algorithms . . . . .	20
4.5	Local Search Computational Complexity . . . . .	23
4.5.1	Score Function Decomposability . . . . .	24
4.6	Summary: What is the Goal . . . . .	24

# Chapter 1

## Introduction

How to learn or estimate the Bayesian network structure from data?

- To find a P-map or minimal I-map structure from data.
- This is called structure learning.

### 1.1 Minimal I-map and P-map

- An I-map is a DAG  $G$ , such that  $I(G) \in I(P)$ .
  - It always exists a fully connected DAG.
- A Minimal I-map is an I-map that if one of its edges is removed, is no longer an I-map.
  - This also always exist. (Just keep removing edges from an I-map)
- A P-map is a DAG  $G$  such that  $I(G) = I(P)$ 
  - A P-map may not exist. The XOR example.
- The best structure is the P-map as it captures all conditional independencies  $I(P)$

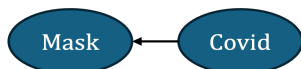
### 1.2 Covid Example

- How to learn the structure in the Covid-Mask problem?

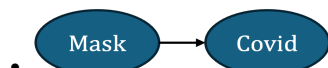
$$P(C, M) = P(C)P(M)$$



$$P(C, M) = P(C)P(M|C)$$



$$P(C, M) = P(M)P(C|M)$$



### 1.2.1 Method 1: Check Independencies

- One approach is to find the data, the conditional independencies that the DAG must satisfies, **the constraints**.

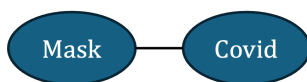
	P(C,M)	P(C)P(M)
$c^0, m^0$	1/12	2/9
$c^0, m^1$	7/12	4/9
$c^1, m^0$	3/12	1/9
$c^1, m^1$	1/12	2/9

#	Covid	Mask
1	1	0
2	0	1
3	1	0
4	0	1
5	0	1
6	0	1
7	1	0
8	0	1
9	0	1
10	0	0
11	0	1
12	1	1

- $P(C, M) \neq P(C)P(M)$ , C and M are dependent.  $I(P) = \emptyset$
- So, the graph is



- PDAG (I-equivalence class)



- This approach is called **constraint based**.
- However, check each structure to see if it satisfies the constraints we found at the table. But if we have a huge network with a large number of variables, check each structure would be very costly.

### 1.2.2 Method 2: Maximizing the Likelihood

- The second approach is based on the idea that the true BN must maximize the likelihood.
- So we find the structure that maximizes the likelihood.

#### Likelihood Approach

- If we have the structure G, we can find the parameters  $\theta_G^{MLE}$  that maximize the probability of observing the data (the likelihood).

- Proposition: MLE parameters: The MLE of the CPD parameters  $\theta_{x_i|pa_{x_i}}$  is obtained by  $\theta_{x_i|pa_{x_i}}^{MLE} = \frac{N[x_i, pa_{x_i}]}{N[pa_{x_i}]}$
- So to maximize the likelihood over all structures, we should find structure G for which parameters  $\theta_G^{MLE}$  maximize the likelihood.

$$\max_G \max_{\theta} L(\theta_G, G|D) = \max_G L(\theta_G^{MLE}, G|D) \quad (1.1)$$

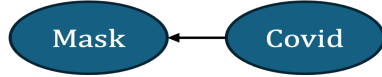
- The second approach is based on the idea that the true BN must maximize the likelihood.
- So we find the structure that maximizes the likelihood.

– Structure 1



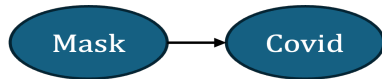
$$\begin{aligned} L(\theta, G_{(M,C)}|D) &= \theta_C^{N[c^1]}(1 - \theta_c)^{N[c^0]} \theta_M^{N[m^1]}(1 - \theta_M)^{N[m^0]} \\ \Rightarrow L(\theta_{(M,C)}^{MLE}, G_{(M,C)}|D) &= \left(\frac{1}{3}\right)^4 \left(\frac{2}{3}\right)^8 \left(\frac{2}{3}\right)^8 \left(\frac{1}{3}\right)^4 = 2.3 * 10^{-7} \end{aligned} \quad (1.2)$$

– Structure 2



$$\begin{aligned} L(\theta, G_{(C \rightarrow M)}|D) &= \theta_C^{N[c^1]}(1 - \theta_c)^{N[c^0]} * \theta_{m^0|c^0}^{N[m^0, c^0]}(1 - \theta_{m^0|c^0})^{N[m^1, c^0]} \\ &\quad * \theta_{m^0|c^1}^{N[m^0, c^1]}(1 - \theta_{m^0|c^1})^{N[m^1, c^1]} \\ \Rightarrow L(\theta_{(C \rightarrow M)}^{MLE}, G_{(C \rightarrow M)}|D) &= \left(\frac{1}{3}\right)^4 \left(\frac{2}{3}\right)^8 \left(\frac{1}{8}\right)^1 \left(\frac{7}{8}\right)^7 \left(\frac{3}{4}\right)^3 \left(\frac{1}{4}\right)^1 = 2.49 * 10^{-6} \end{aligned} \quad (1.3)$$

– Structure 3



$$\begin{aligned} L(\theta, G_{(M \rightarrow C)}|D) &= \theta_M^{N[m^1]}(1 - \theta_M)^{N[m^0]} * \theta_{c^0|m^0}^{N[m^0, c^0]}(1 - \theta_{c^0|m^0})^{N[m^0, c^1]} \\ &\quad * \theta_{c^0|m^1}^{N[m^1, c^0]}(1 - \theta_{c^0|m^1})^{N[m^1, c^1]} \\ \Rightarrow L(\theta_{(M \rightarrow C)}^{MLE}, G_{(M \rightarrow C)}|D) &= \left(\frac{2}{3}\right)^8 \left(\frac{1}{3}\right)^4 \left(\frac{1}{4}\right)^1 \left(\frac{1}{4}\right)^3 \left(\frac{7}{8}\right)^7 \left(\frac{1}{8}\right)^1 = 2.49 * 10^{-6} \end{aligned} \quad (1.4)$$

– Hence,

$$\begin{aligned} \max_G \max_{\theta} L(\theta_G, G|D) &= L(\theta_{(C \rightarrow M)}^{MLE}, G_{(C \rightarrow M)}|D) \\ &= L(\theta_{(M \rightarrow C)}^{MLE}, G_{(M \rightarrow C)}|D) \end{aligned} \quad (1.5)$$

– So the class PDAG is



– The likelihood is a score to measure goodness of fit. This approach is called **score based**.

## 1.3 Structure Learning Approaches

Two main approaches to learn the structure are

- Constraint based
  - requires finding  $I(P)$  from data
  - an algorithm to find the structure from  $I(P)$
- Score based
  - requires a score function such as the likelihood
  - an algorithm to find the structure that maximizes the score.

## 1.4 Summary: What is the Goal?

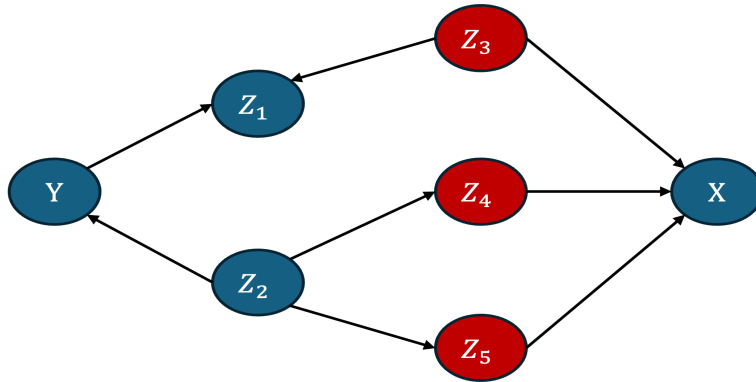
- The goal was to find a BN that factorize the joint distribution according to all of the conditional independencies in the data.
- This is equivalent to the structure being a P-map if one exists; otherwise only a minimal I-map is possible.
- So, the goal was to find a P-map (or a minimal I-map) or the class PDAG as some links can be arbitrary directions.
- If we exhaustively check all possible structures, it is computational costly.

## Chapter 2

# Constraint-based Algorithm

### 2.1 From Distribution to Graph (Finding P-map)

- Non-existence of a link implies some conditional independence.
- For instance, in the following network, there is no link between X and Y, and we have  $X \perp Y | P_{a_X}$



- Lemma: Let G be a P-map for P. There is no link between X and Y in G, if and only if either

$$P \models (X \perp Y | P_{a_X}), P \models (X \perp Y | P_{a_Y}) \quad (2.1)$$

- This leads to an algorithm to obtain the skeleton of the P-map.

### 2.2 Finding the P-map Skeleton

Algorithm: Recovering the undirected skeleton of the P-map for P.

- Construct the completely connected undirected graph H.
- Repeatedly remove the edge between any pair of nodes X, Y in H that are independent conditioned on a subset of either of their exclusive neighbors. That is to remove the edges X - Y if

$$P \models (X \perp Y | U_{X,Y}) \quad (2.2)$$

for some  $U_{X,Y} \subseteq N_X^H$  or  $U_{X,Y} \subseteq N_Y^H$ , where  $N_X^H$  is the neighbor set of X in H.

- The final graph is the skeleton of the P-map.



- $U_{X,Y}$  is potentially  $P_{a_X}^G$  or  $P_{a_Y}^G$  in the lemma.
- To speed up the process,  $U$  can be checked from small to large subsets. first,  $|U| = 0$ , then  $|U| = 1$ , etc.

## 2.3 The Covid Problem

Find the P-map for mask, social distancing, covid, difficulty breathing, fever, and ventilation.

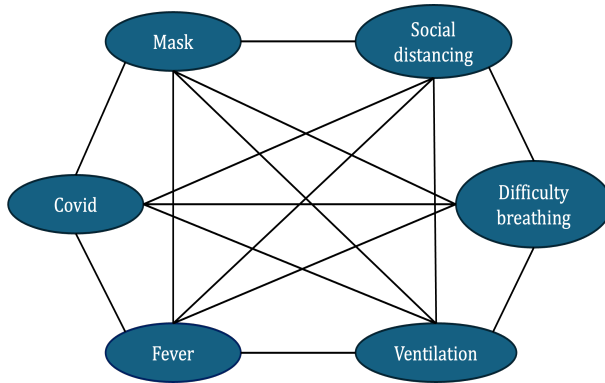
#	Covid	Mask	Social Distancing	FlU	COugh	Fever	Ventilation	Season	ConGestion	Difficulty Breathing	DRug	Allergy
1	1	0	1	0	1	1	1	Spring	1	1	0	0
2	0	1	1	0	0	1	0	Summer	0	0	1	0
3	1	0	0	1	1	0	0	Fall	0	1	1	0
4	0	1	1	0	0	1	0	Winter	1	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...
1000	1	1	1	0	1	0	1	Spring	0	0	1	1

Assume that following conditional independencies can be obtained from the table:

$$\begin{aligned}
&1. (S \perp M), (M \perp D), (S \perp D), (S \perp M, D), (D \perp S, M), (M \perp S, D) \\
&2. (A \perp M, D, U|S), (U \perp M, D, A|S) \\
&3. (C \perp S, A, U|M, D) \\
&4. (G \perp S, M, D, C, F, O, B, R, V|A, U) \\
&5. (F \perp S, M, D, A, G, O, B, V|U, C), (O \perp S, M, D, A, F, G, B, V|U, C) \\
&6. (B \perp S, M, D, A, U, G, F, O|C) \\
&7. (R \perp S, M, D, A, U, C, G, B, V|F, O) \\
&8. (V \perp S, M, S, A, U, C, G, F, O, R|B)
\end{aligned} \tag{2.3}$$

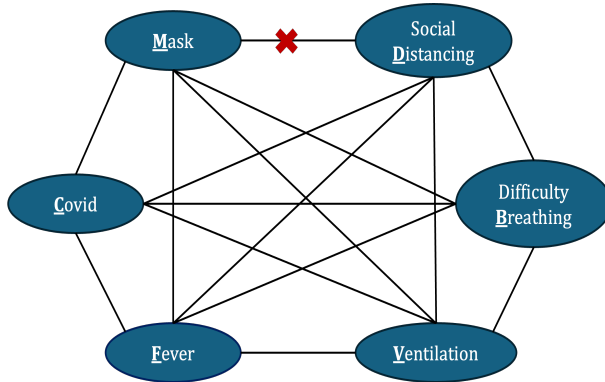
**Step 1: Construct the completely connected undirected graph over the variables**

.

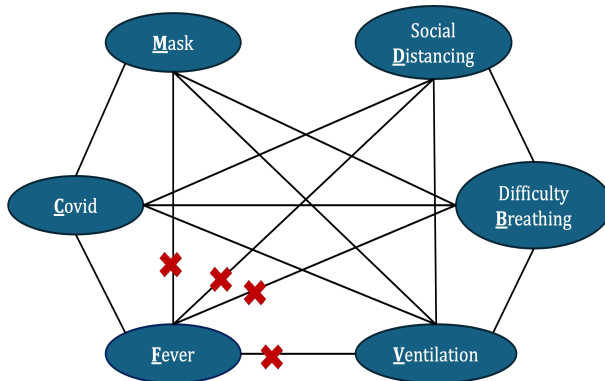


**Step 2: Find the skeleton: Repeatedly remove the edge between any pair of nodes X, Y that are independent conditioned on a subset of either of their exclusive neighbors**

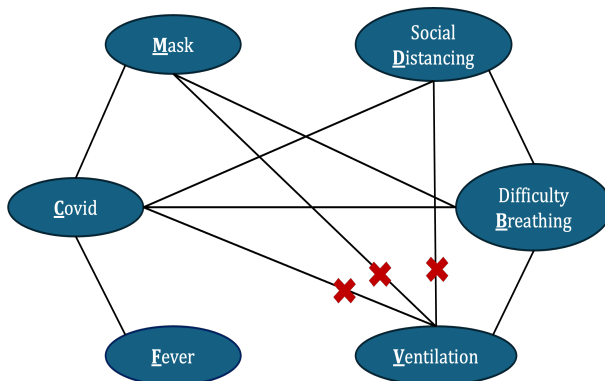
- From  $(M \perp D)$ , social distancing is independent of mask, so the edge connecting them is removed.



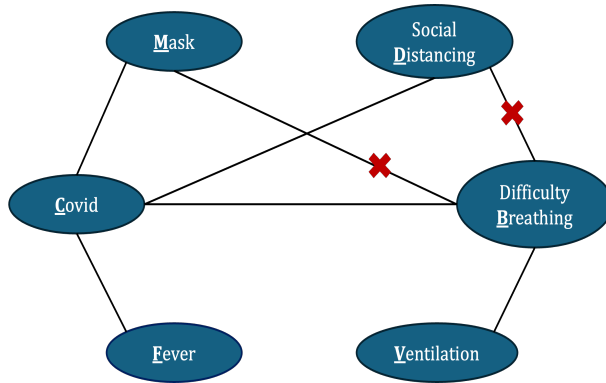
- From  $(F \perp M, D, B, V|C)$ , given covid, fever is independent of all the other variables; thus, the corresponding edges are removed.



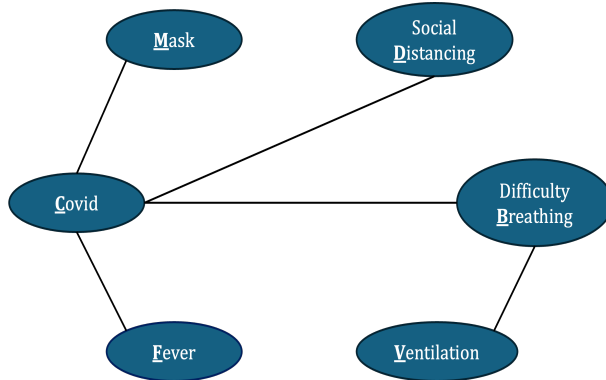
- From  $(V \perp M, D, C|B)$ , given difficulty breathing, ventilation is independent of covid, mask, social distancing.



- From  $(B \perp M, D|C)$ , given covid, difficulty breathing is independent of mask and social distancing.

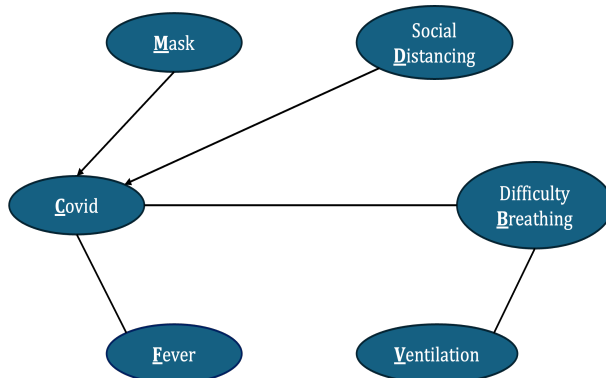


- Eventually, we have the following skeleton.

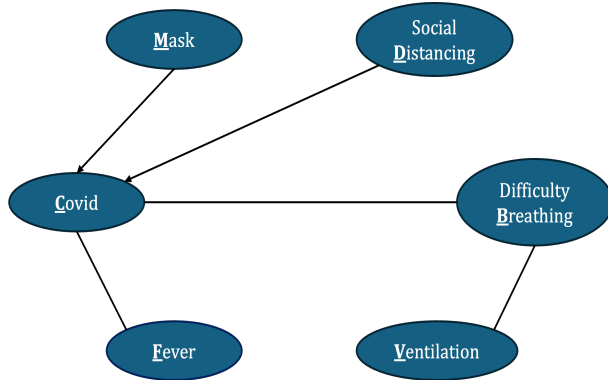


### Step 3: Determine the directions of the edges.

- Proposition: Let  $G$  be a P-map for  $P$  and  $X - Z - Y$  be a connected triple without an edge between  $X$  and  $Y$ . If  $X \rightarrow Z \leftarrow Y$  belong to  $G$ , i.e. an immorality. then,  $P \not\models (X \perp Y | U)$  for all set  $U$  that contains  $Z$ .
- So, if  $Z$  belongs to the  $U$  in the previous algorithm, it must not form a v-structure with  $X$  and  $Y$ .
- Proposition: Let  $G$  be a P-map for  $P$  and  $X - Z - Y$  be a connected triple without an edge between  $X$  and  $Y$ . If  $X \rightarrow Z \leftarrow Y$  does not belong to  $G$ . then,  $P \models (X \perp Y | U)$  implies  $U$  contains  $Z$ .
- So, if  $Z$  does not belong to the  $U$  in the previous algorithm, it must form a v-structure with  $X$  and  $Y$ .
- From  $M \perp D$ , we could have  $C \notin U_{M,D} = \emptyset$ , so we form the v-structure.



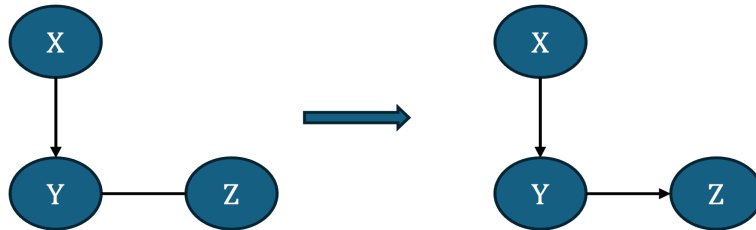
- For each connected triple,  $X-Z-Y$ , where  $X$  and  $Y$  are not connected, if  $Z \in U_{X,Y}$ , then do nothing.  
Given  $(V \perp S, M, D, A, U, C, G, F, O, R|B)$ 
  - For D-C-B, we have  $(B \perp D|C)$ , thus  $C \in U_{B,C}$
  - For D-C-F, we have  $(F \perp D|C)$ , thus  $C \in U_{F,D}$
  - For B-C-F, we have  $(F \perp B|C)$ , thus  $C \in U_{F,B}$
  - For C-B-V, we have  $(C \perp V|B)$ , thus  $B \in U_{F,V}$
- Eventually, we have



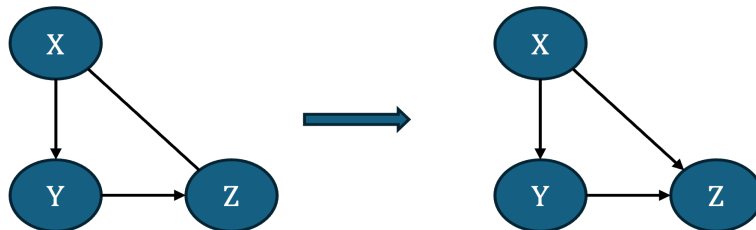
#### Step 4: Determine Remaining Edges

Orienting remaining edges: avoid directed cycles and new immoralities.

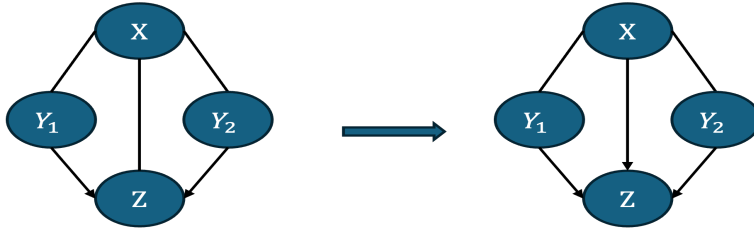
- Rule r1



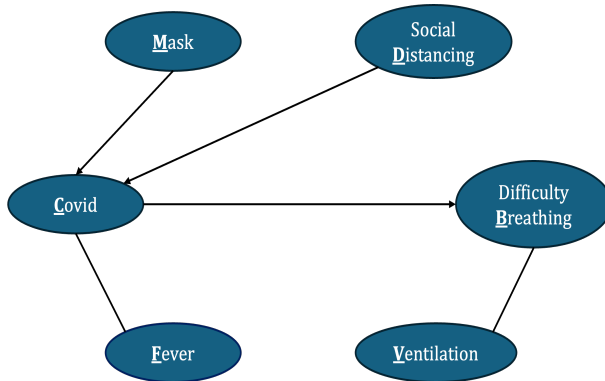
- Rule r2



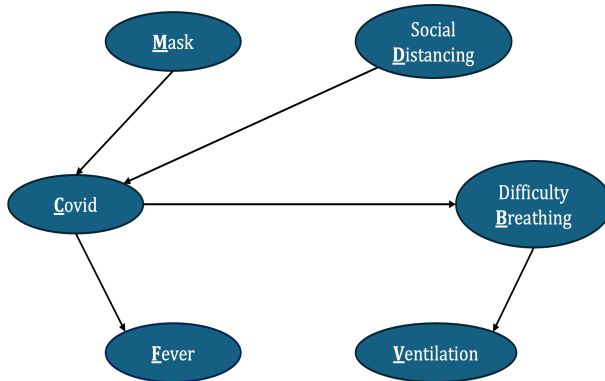
- Rule r3



- For C-B setting direction toward C creates immorality (R1) so we set it in opposite direction.



- For C-F setting direction toward F to avoid immorality (R1). For B-V setting direction toward V to avoid immorality (R1)



## 2.4 Finding PDAG

### 2.4.1 Algorithm: Discovering the class PDAG of a P-map of a Distribution

- Algorithm:
  - Obtain the skeleton (step 1)
  - Find the immoralities: for each connected triple  $X-Z-Y$ , where  $X$  and  $Y$  are not connected, if  $Z \notin U_{X,Y}$ , then add the directions  $X \rightarrow Z \leftarrow Y$ .
  - Find the orientations of the other edges by obeying the R1, R2, R3.
- This is known as the PC algorithm.
- Let  $P$  be a distribution that has a P-map  $G$ . The graph obtained from the above procedure is a class PDAG of  $G$ .

### 2.4.2 Order of Complexity

- To assess the efficiency of an algorithm, we evaluate its computational time complexity.
- Computational time complexity is a measure of the algorithm run time based on its input size.
- For example, if the input is a list, the number of elements in that list can be the input size.
- If the input size is  $n$ , the computational time complexity is defined as a function  $f(n)$ .
- We say an algorithm is of order  $O(g(n))$  and write  $f(n) = O(g(n))$ , if there exists a constant  $c$  such that for each input of size  $n$ , we have  $f(n) \leq cg(n)$ .
- Time complexity of the PC algorithm:  $O(n^{d+2})$ , when the number of parents is limited to  $d$ . i.e.  $P_{a_x} < d$  for all  $X$ .

## 2.5 Summary: What is the Goal

- The goal is to systematically find a class PDAG P-map from data.
- One approach is constraint based
  - Starting from a fully connected graph, iteratively remove edges between every pair of nodes  $X, Y$  that are conditionally independent given one or more of their neighbors.  $N_x$  - the parents of  $X$  or  $Y$ .
  - Turn  $X, Y$ , and every non-parent neighbor into a v-structure.
  - Orient other edges to avoid cycles and new immoralities.

## Chapter 3

# Finding Independencies from Data

### 3.1 Independence Test: Null Hypothesis Test

How to check a conditional independence from the data?

- For example, are covid and mask independent of each other?
- In practice, due to noise, we cannot check if the following equality perfectly holds.

$$P(C, M) = P(C)P(M) \quad (3.1)$$

- A statistical method should be used.
- We define Null Hypothesis  $H_0$  as

$$P(C, M) = P(C)P(M) \quad (3.2)$$

and use statistic tests to verify or falsify it.

- For example, if  $H_0$  is true, we expect the number of instances with  $C = 0$  and  $M = 0$ . i.e.  $N[c^0, m^0]$  to be  $NP(c^0)P(m^0)$ , where  $N$  is the total number of samples.
- To test the hypothesis, we can compare the observed number of samples in the data with this expected value.
- Then the Chi-square(d) statistic is

$$x^2(D) = \sum_{i \in V} \frac{(O_i - E_i)^2}{E_i} \quad (3.3)$$

- If it is large enough, the hypothesis is rejected.

### 3.2 Chi-squared Distribution

- If  $H_0$  is true, the mismatches should be due to random errors.
- Then  $Z_i = (Q_i - E_i)/\sqrt{E_i}$  can be considered as independent random variables that follow a standard normal distribution?

$$Q = \sum_{i=1}^k Z_i^2, Z_i \sim N(0, 1) \quad (3.4)$$

- This is almost true, just that  $Z_i = (Q_i - E_i)/\sqrt{E_i}$  are not independent. so  $Z_i$  is take a different form.
- The resulting distribution is called the **Chi-squared distribution** with degree of freedom (df)  $k$  (number of independent pieces of information).
- For two variables  $X$  and  $Y$ ,

$$df = (|Val(x)| - 1) * (|Val(y)| - 1) \quad (3.5)$$

## Level of Significance

- If the computed chi-squared statistic has a low probability in the distribution,  $H_0$  is rejected. What is low?
- We define the level of significance  $\alpha$  to be

$$\alpha \triangleq P(\text{rejected } H_0 | H_0 \text{ is valid}) \quad (3.6)$$

- The  $\alpha$  level represents the accepted probability of making a type I error where  $H_0$  is falsely rejected.
- So  $\alpha = 0.05$  means that it is fine to falsely reject  $H_0$  5% of the time.
- We reject  $H_0$  only if the chi-squared statistic is extreme enough to fall in the rejection region.

## P-value

- The p-value is the probability of observing values equal to or more extreme than the computed Chi-squared statistic:  $P(Q \geq X^2(D))$
- If p-value  $< \alpha$ , then  $H_0$  is rejected with confidence  $1 - \alpha$
- This is equivalent of having a chi-squared statistic  $X^2(D)$  greater than the critical value  $X_\alpha^2$ , i.e.

$$X^2(D) > X_\alpha^2 \Rightarrow H_0 \text{ rejected} \quad (3.7)$$

## Covid Example

In the covid-mask problem,

- Null hypothesis ( $H_0$ ):  $C \perp M$  or  $P(C, M) = P(C)P(M)$
- Observed count:  $N[c, m]$
- Expected count number:  $H_0 = NP(c, m) = NP(c)P(m) = \frac{N[c]N[m]}{N}$
- $V = \{(c^0, m^0), (c^0, m^1), (c^1, m^0), (c^1, m^1)\}$

$$\begin{aligned} X^2(D) &= \sum_{c,m} \frac{(N[c, m] - NP(c)P(m))^2}{NP(c)P(m)} \\ &= \sum_{c,m} \frac{(N[c, m]N - P(c)P(m))^2}{NP(c)P(m)} \\ &= \frac{(1 * 12 - 8 * 4)^2}{8 * 4 * 12} + \frac{(7 * 12 - 8 * 8)^2}{8 * 8 * 12} \\ &\quad + \frac{(3 * 12 - 4 * 4)^2}{4 * 4 * 12} + \frac{(1 * 12 - 4 * 8)^2}{4 * 8 * 12} \end{aligned} \quad (3.8)$$



- We have two variables with two values. So,  $df = (\text{Number of value of covid} - 1) * (\text{Number of values of mask} - 1) = (2 - 1) * (2 - 1) = 1$
- Let  $\alpha = 0.05$ , then the critical chi-squared value  $X_{\alpha}^2$  is 3.84

$$X^2(D) = 4.69 > X_{0.05}^2 = 3.84 \quad (3.9)$$

- Therefore,  $p\text{-value} < \alpha$ , null hypothesis  $H_0$  is rejected. Covid and Mask are dependent.

### 3.3 Independence Test for 3 Variables

- To investigate the conditional independence ( $X \perp Y|Z$ ), the null hypothesis  $H_0$  is

$$P(X, Y, Z) = P(Z)P(X|Z)P(Y|Z) \quad (3.10)$$

- We can use the Chi-squared test.
- The observed count is  $O = N[x, y, z]$
- The expected count is

$$\begin{aligned} E = NP(x, y, z) &= NP(z)P(x|z)P(y|z) \\ &= N \frac{N[z]}{N} \frac{N[x, z]}{N[z]} \frac{N[y, z]}{N[z]} \\ &= \frac{N[x, z]N[y, z]}{N[z]} \end{aligned} \quad (3.11)$$

- So we have,

$$\begin{aligned} X^2(D) &= \sum_z \sum_{x,y} \frac{(N[x, y, z] - NP(z)P(x|z)P(y|z))^2}{NP(z)P(x|z)P(y|z)} \\ &= \sum_z \sum_{x,y} \frac{(N[x, y, z]N[z] - N[x, z]N[y, z])^2}{N[z]N[x, z]N[y, z]} \end{aligned} \quad (3.12)$$

- The degree of freedom is  $|\text{val}(z)| * |\text{val}(x) - 1| * |\text{val}(y) - 1|$

### 3.4 Summary: What is the Goal

- The goal is to systematically find a class PDAG P-map from data.
- One approach is constraint based
  - Starting from a fully connected graph, iteratively remove edges between every pair of nodes  $X, Y$  that are conditionally independent given one or more of their neighbors.  $N_x$  - the parents of  $X$  or  $Y$ .
    - \* how to know if they are conditionally independent? using the **chi-squared test**.
  - Turn  $X, Y$ , and every non-parent neighbor into a v-structure.
  - Orient other edges to avoid cycles and new immoralities.

## Chapter 4

# Score-based Algorithm

In the score-based approach, we search for the structure with the highest score.

### 4.1 The Likelihood Score

- Definition: the likelihood score of a DAG  $G$  is defined as

$$score_L(G; D) = l(\theta_G^{MLE} | D) = \log L(\theta_G^{MLE} | D) \quad (4.1)$$

where  $\theta_G^{MLE}$  is the maximum likelihood estimate of  $G$ 's parameters.

$$\theta_G^{MLE} = \operatorname{argmax}_{\theta} l(\theta_G | D) \quad (4.2)$$

- Log can take any base although it is common to consider the base  $e$ .

#### 4.1.1 Recall: Entropy and Mutual Information

- Consider random variables  $X$  and  $Y$ .
- The entropy  $H_p(X)$  is a measure of randomness and uncertainty in the values of  $X$ :

$$H_p(X) = - \sum_x P(x) \log P(x) \quad (4.3)$$

- The higher the entropy, the more uncertain the outcome is.
- The mutual information  $I_p(X, Y)$  measures the dependency between  $X$  and  $Y$ .

$$I_p(X, Y) = \sum_{x,y} P(x, y) \log \left( \frac{P(x, y)}{P(x)P(y)} \right) \quad (4.4)$$

- Mutual information is non-negative
- A mutual information of zero implies independence.

### 4.1.2 Likelihood Score Proposition

- Proposition: The likelihood score decomposes as follows:

$$score_L(G; D) = N \sum_{i=1}^N I_{\hat{P}}(X_i, P_{a_{X_i}}) - N \sum_{i=1}^n H_{\hat{P}}(X_i) \quad (4.5)$$

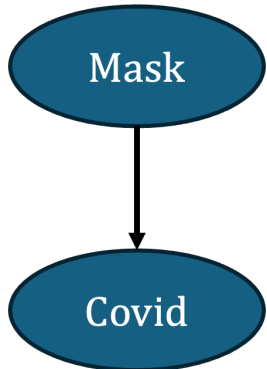
where  $\hat{P}$  is the empirical distribution. i.e.  $\hat{P}(x)$  is the frequency of  $X$  in  $D$ .

- $\sum_{i=1}^N I_{\hat{P}}(X_i, P_{a_{X_i}})$  mutual information between each variable and its parents.
- $N \sum_{i=1}^n H_{\hat{P}}(X_i)$  entropy of all variables.
- Proof omitted

### 4.1.3 The Covid Example

- Calculate the likelihood score of the following structure.

$$\begin{aligned} score_L(G, D) &= 12 * (I_{\hat{P}}(C, P_{a_C})) - 12 * (H_{\hat{P}}(C) + H_{\hat{P}}(M)) \\ &= (12 * 0.29) - (12 * 1.84) = -18.6 \end{aligned} \quad (4.6)$$



#	Covid	Mask
1	1	0
2	0	1
3	1	0
4	0	1
5	0	1
6	0	1
7	1	0
8	0	1
9	0	1
10	0	0
11	0	1
12	1	1

$$\begin{aligned} H_{\hat{P}}(M) &= -\hat{P}(M=1) \log \hat{P}(M=1) - \hat{P}(M=0) \log \hat{P}(M=0) \\ &= \frac{8}{12} * \log\left(\frac{12}{8}\right) + \frac{4}{12} * \log\left(\frac{12}{4}\right) = 0.92 \\ H_{\hat{P}}(C) &= -\hat{P}(C=1) \log \hat{P}(C=1) - \hat{P}(C=0) \log \hat{P}(C=0) \\ &= \frac{4}{12} * \log\left(\frac{12}{4}\right) + \frac{8}{12} * \log\left(\frac{12}{8}\right) = 0.92 \\ I_{\hat{P}}(C, P_{a_C}) &= \hat{P}(C=1, M=1) \log\left(\frac{\hat{P}(C=1, M=1)}{\hat{P}(C=1) \hat{P}(M=1)}\right) \\ &\quad + \hat{P}(C=1, M=0) \log\left(\frac{\hat{P}(C=1, M=0)}{\hat{P}(C=1) \hat{P}(M=0)}\right) \\ &\quad + \hat{P}(C=0, M=1) \log\left(\frac{\hat{P}(C=0, M=1)}{\hat{P}(C=0) \hat{P}(M=1)}\right) \\ &\quad + \hat{P}(C=0, M=0) \log\left(\frac{\hat{P}(C=0, M=0)}{\hat{P}(C=0) \hat{P}(M=0)}\right) = 0.29 \end{aligned} \quad (4.7)$$

#### 4.1.4 Interpretation: Maximizing Dependence on Parents

$$score_L(G, D) = N \sum_{i=1}^n I_{\hat{P}}(X_i, Pa_{X_i}) - N \sum_{i=1}^n H_{\hat{P}}(X_i) \quad (4.8)$$

- The entropy term  $H_{\hat{P}}(X_i)$  does not depend on the network structure.
- Thus, different network vary only in their mutual information terms.
- Mutual information  $I_{\hat{P}}(X_i, Pa_{X_i})$  represents the strength of the dependence between  $X_i$  and its parents  $Pa_{X_i}$  in  $P$ .
- Higher score networks encode greater parental dependencies.

$$I_{\hat{P}}(X_i, Pa_{X_i} \cup Y) \geq I_{\hat{P}}(X_i, Pa_{X_i}) \quad (4.9)$$

- So adding another parent almost always increase the mutual information between the variable and its parents.
- So the  $score_L(G; D)$  is maximized for a completely connected network.
- This likelihood score yields a model that **overfits** the training data. It may not generalize well to new datasets.

## 4.2 The Bayesian Score

- The Bayesian approach is to consider a distribution over uncertainties, which here are the structure  $G$  and its parameters  $\theta_G$ , so we define
  - the structure prior  $P(G)$
  - the parameter priors of a given structure  $G$ , i.e.  $P(\theta_G|G)$ .
- Using the Bayes rule, the probability of a graph  $G$  given data  $D$  is

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)} \quad (4.10)$$

- The Bayesian score is defined accordingly

$$score_B(G, D) = \log P(D|G) + \log P(G) \quad (4.11)$$

- $P(D)$  does not depend on the structure and is excluded.
- $P(D|G)$  is the **marginal likelihood** of the data given the structure.

$$P(D|G) = \int P(D|\theta_G, G)P(\theta_G|G)d\theta_G \quad (4.12)$$

- Proposition: given the parameter prior  $P(\theta_G|G)$  for the structure  $G$  that satisfies the local and global parameter independence.

$$P(D|G) = \prod_{i=1}^n \prod_{pa_{X_i}} \int \prod_{t, pa_{X_i}[t]=pa_{X_i}} P(X_i[t]|pa_{X_i}, \theta_{X_i|pa_{X_i}}, G) P(\theta_{X_i|pa_{X_i}}|G) d\theta_{X_i|pa_{X_i}} \quad (4.13)$$

- Proposition: If the parameter prior distribution of a Bayesian network satisfies  $P(\theta_{X_i|pa_{X_i}}|G) = \text{Dirichlet}(\alpha_{X_i^1|pa_{X_i}}, \dots, \alpha_{X_i^k|pa_{X_i}})$ , then given the dataset  $D = \xi[1], \dots, \xi[N]$ , the marginal likelihood is

$$P(D|G) = \prod_{i=1}^n \prod_{pa_{X_i}} \frac{\Gamma(\sum_k \alpha_{X_i^k|pa_{X_i}})}{\Gamma(\sum_k \alpha_{X_i^k|pa_{X_i}} + N[pa_{X_i}])} \sum_k \frac{\Gamma(\alpha_{X_i^k|pa_{X_i}} + N[X_i^k, pa_{X_i}])}{\Gamma(\alpha_{X_i^k|pa_{X_i}})} \quad (4.14)$$

### 4.3 The BIC Score

- Define the **model dimension**  $\text{Dim}[G]$  as the number of independent parameter in DAG  $G$ , capturing model complexity.
- Proposition: given a Dirichlet parameter prior  $P(\theta_G|G)$  as  $N \rightarrow \infty$

$$\log P(D|G) = l(\theta_G^{MLE}|D) - \frac{1}{2} \log N * \text{Dim}[G] + O(1) \quad (4.15)$$

- The log-likelihood term  $l(\theta_G^{MLE}|D)$  measures fit to data, and the second term measures model complexity.
- So, the approximation trades off between data fitting and model complexity.
- The **Bayesian Information Criterion (BIC) Score**

$$\text{Score}_{BIC}(G; D) = l(\theta_G^{MLE}|D) - \frac{1}{2} \log N * \text{Dim}[G] \quad (4.16)$$

- By substituting the expression for the likelihood score, we obtain

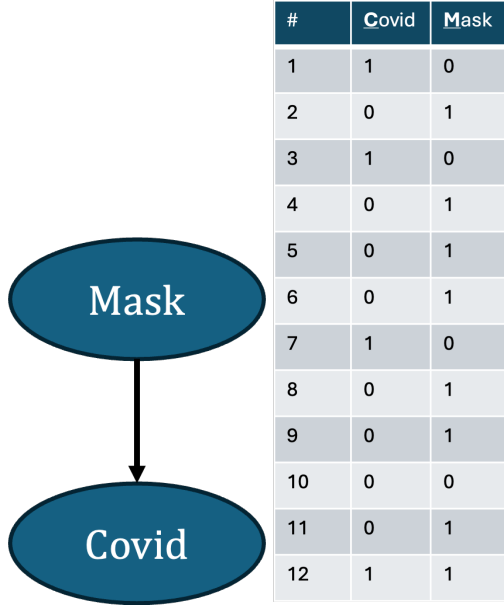
$$\text{Score}_{BIC}(G; D) = N \sum_{i=1}^n I_{\hat{P}}(X_i, P_{a_{X_i}}) - N \sum_{i=1}^n H_{\hat{P}}(X_i) - \frac{1}{2} \log N * \text{Dim}[G] \quad (4.17)$$

- The entropy terms are irrelevant to the structure.
- Mutual information grows linearly whereas model complexity grows logarithmically with the data size.
- So for small datasets, simpler models are preferred.
- As the size of the dataset increases, so does the preference for more complex models.

#### 4.3.1 The Covid Example

- To obtain the BIC, we can subtract the penalty term from the likelihood score.

$$\begin{aligned} \text{score}_{BIC}(G, D) &= 12 * (I_p(C, P_{a_C})) - 12 * (H_p(C) + H_p(M)) - \frac{\log N}{2} * \text{Dim}[G] \\ &= (12 * 0.29) - (12 * 1.84) - \frac{\log 12}{2} * 3 = -23.98 \end{aligned} \quad (4.18)$$

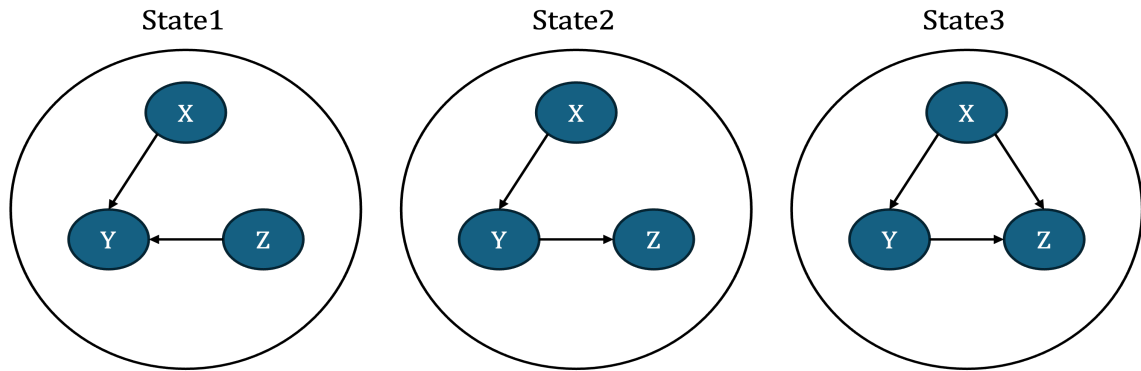


$$\begin{aligned}
H_{\hat{P}}(M) &= -\hat{P}(M=1) \log \hat{P}(M=1) - \hat{P}(M=0) \log \hat{P}(M=0) \\
&= \frac{8}{12} * \log \frac{12}{8} + \frac{4}{12} * \log \frac{12}{4} = 0.92 \\
H_{\hat{P}}(C) &= -\hat{P}(C=1) \log \hat{P}(C=1) - \hat{P}(C=0) \log \hat{P}(C=0) \\
&= \frac{4}{12} * \log \frac{12}{4} + \frac{8}{12} * \log \frac{12}{8} = 0.92 \\
I_{\hat{P}}(C, P_{a_C}) &= \hat{P}(C=1, M=1) \log \left( \frac{\hat{P}(C=1, M=1)}{\hat{P}(C=1) \hat{P}(M=1)} \right) \\
&\quad + \hat{P}(C=1, M=0) \log \left( \frac{\hat{P}(C=1, M=0)}{\hat{P}(C=1) \hat{P}(M=0)} \right) \\
&\quad + \hat{P}(C=0, M=1) \log \left( \frac{\hat{P}(C=0, M=1)}{\hat{P}(C=0) \hat{P}(M=1)} \right) \\
&\quad + \hat{P}(C=0, M=0) \log \left( \frac{\hat{P}(C=0, M=0)}{\hat{P}(C=0) \hat{P}(M=0)} \right) = 0.29
\end{aligned} \tag{4.19}$$

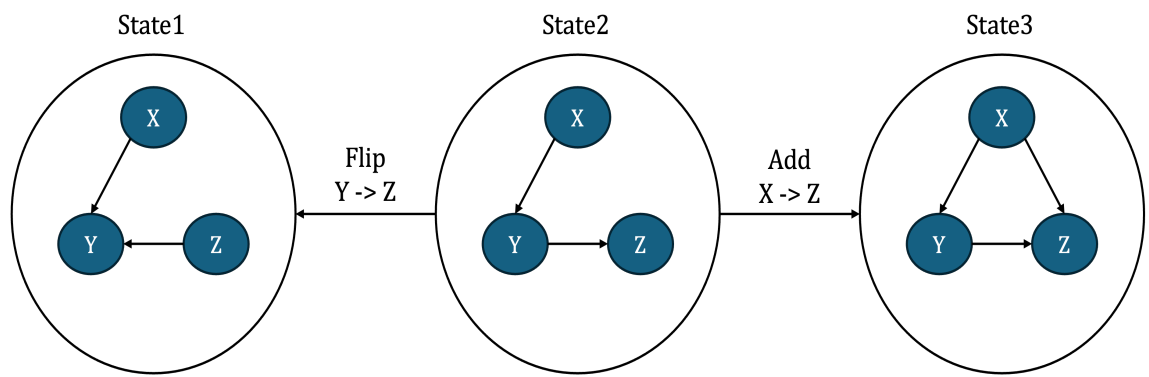
## 4.4 The Learning Procedure

### 4.4.1 Score-based Algorithm

- Score-based algorithms are based on a **score** and a **search procedure**.
- The score measures the goodness of fit of the structure to the data
- An algorithm searches for the structure that maximizes the score. The search space consists of all possible DAGs with the random variables as the nodes.
- Each DAG is a state in the search space.

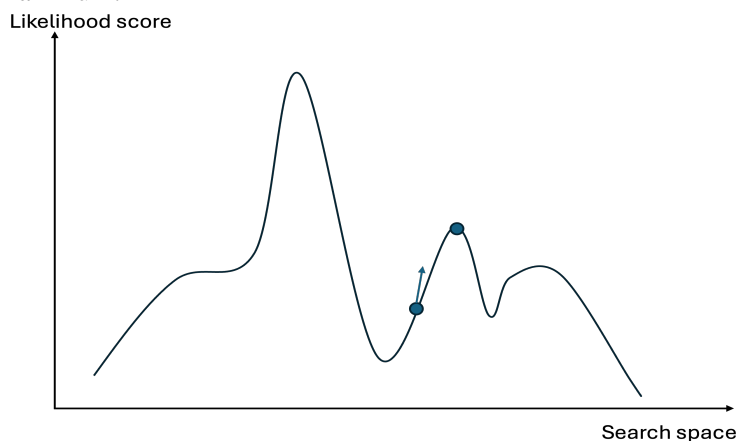


- In a local search procedure, neighboring states are formed.
  - The DAGs obtained by applying an operator on the current DAG.
  - We consider these operators: adding, deleting, or flipping of a link.



#### 4.4.2 Search Procedure: Algorithms

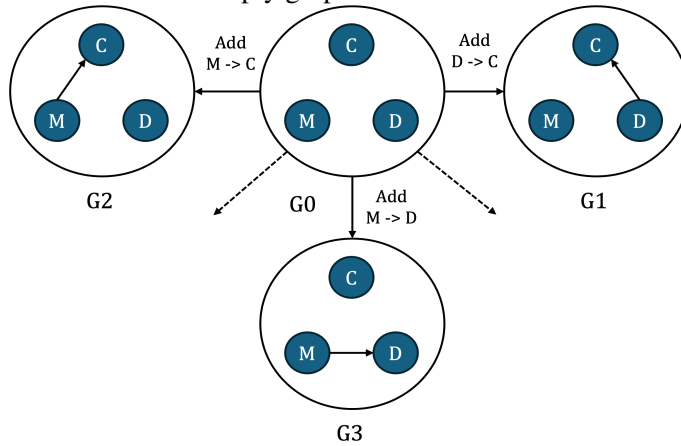
- Starting from the current state, by moving to the neighboring state with the highest likelihood score, a local maximum is reached.
- Local search algorithms such as greedy hill climbing search and tabu search find such a local maximum.



#### Greedy Hill Climbing Search

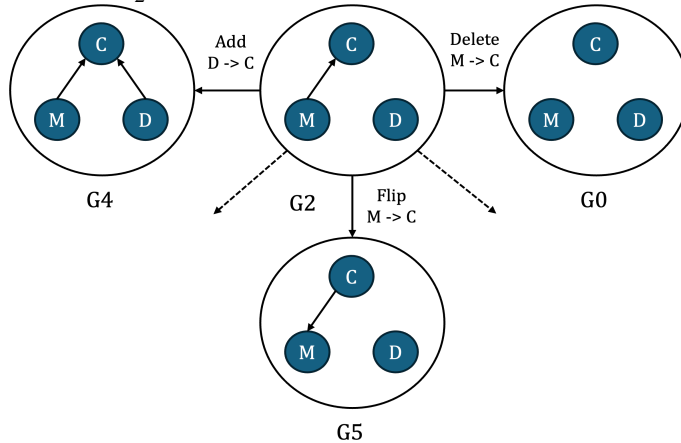
- Greedy hill climbing search:

- Start from an initial state  $G_0$ .
- Find a neighbor for  $G_0$  with the maximum score.
- Repeat this loop to find the local maximum.
- Greedy hill climbing is similar to the continuous gradient ascent.
- The Covid Example:
  - Use greedy hill climbing search with the likelihood score to find the BN structure for covid, mask, and social distancing.
  - We start with an empty graph.



#	Covid	Mask	Distancing
1	0	1	1
2	1	0	0
3	0	1	1
4	1	0	1
5	0	1	0
6	0	1	0
7	0	1	0
8	0	0	0
9	0	1	1
10	1	0	1
11	0	1	1
12	1	0	1

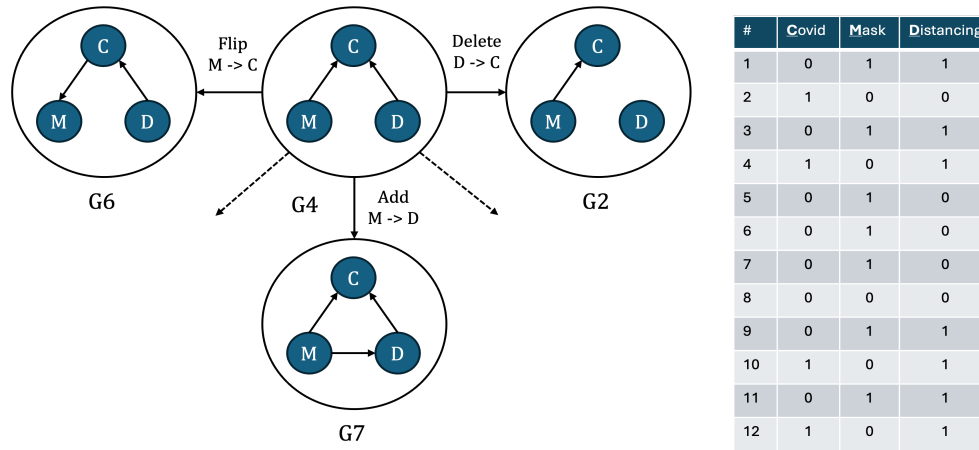
- The scores of the neighbors are  $score_L(G_1) = -10.24$ ,  $score_L(G_2) = -8.17$ ,  $score_L(G_3) = -10.39$ .  $G_2$  has the maximum score so move to this state.



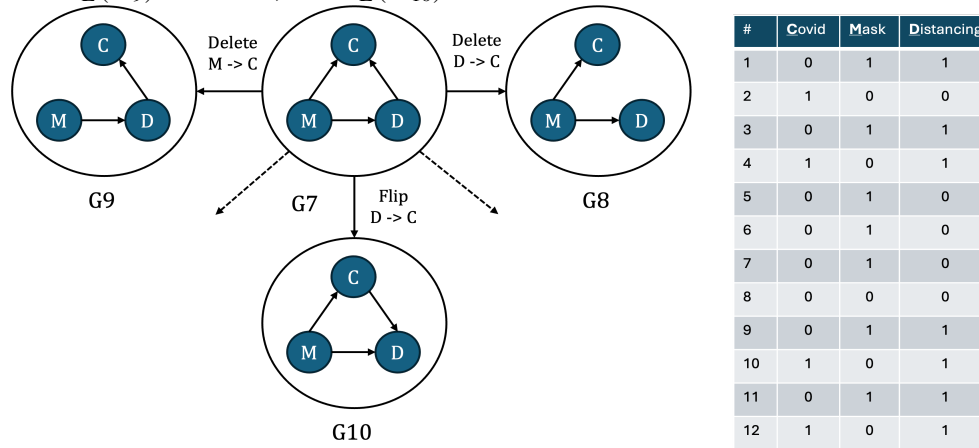
#	Covid	Mask	Distancing
1	0	1	1
2	1	0	0
3	0	1	1
4	1	0	1
5	0	1	0
6	0	1	0
7	0	1	0
8	0	0	0
9	0	1	1
10	1	0	1
11	0	1	1
12	1	0	1

- The scores of the neighbors are  $score_L(G_0) = -10.4$ ,  $score_L(G_4) = -7.68$ ,  $score_L(G_5) = -8.17$ . For the new state,  $G_4$  is the neighbor with the maximum score.





- The score of  $G_7 = -7.68$  is equal to or higher than all of its neighbors.  $score_L(G_8) = -8.16$ ,  $score_L(G_9) = -10.14$ ,  $score_L(G_{10}) = -7.68$ . We reached a local maximum.



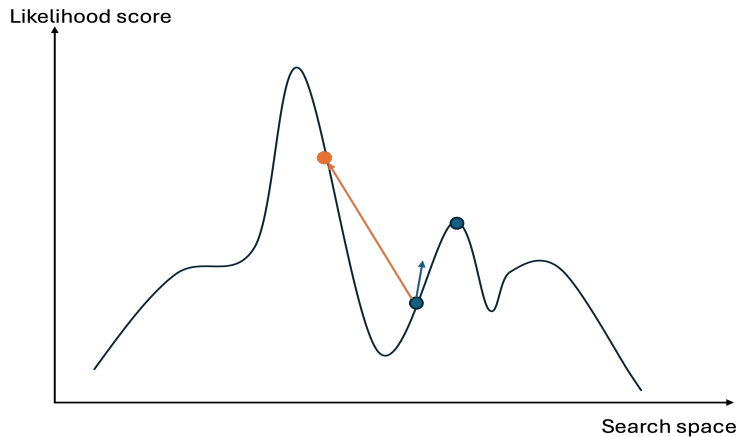
- One issue is that the number of neighbors to search at each node can be too many. One solution is to evaluate only a sample of the neighbors. (first ascent hill climbing).
- As with other local search algorithms, hill climbing is likely to get stuck in a local maximum.

## Tabu Search

- We can force exploring new directions, even if the immediate visited states are not scoring higher.
- Tabu Search: once an operator is applied, do not reverse it for a while.
  - Set an initially empty 'tabu' list of operators. each operator added stays there for K steps and is then removed.
  - Start with an initial state  $G_0$ .
  - Choose an operator whose reverse is not in the tabu list.
  - If the resulting neighbor has a higher score, then add the operator to the tabu list, then add the operator to the tabu list and hold it there for the next k steps. then, move to the resulting neighbor.
  - Increase the step number by one.
  - Stop if the desired number of improvements is reached. otherwise, go to step 3.

## Random Walk

- Another approach is to occasionally move to a randomly selected non-neighbor state.
- This is the random walk approach and increases the chance of reaching the global maximum.



## Computational Complexity

- None of these methods guarantee a global maximum.
- A global search usually requires the whole search space to be searched, which is computationally costly.
- Even for the case where the score function is decomposable and the maximum number of parameters of a node is limited to  $d$ , finding the highest-score DAG is an NP-hard problem.
- So, usually we have to rely on a local search.

## 4.5 Local Search Computational Complexity

- Suppose that it takes  $K$  steps to reach the highest score DAG  $G^*$  from the initial DAG  $G_0$ .
- How many operators are applied and evaluated at each step? i.e. finding the neighboring DAG with the highest score.
  - The maximum number of edges in a DAG with  $n$  vertices is  $\frac{n(n-1)}{2} = O(n^2)$
  - For each edge, either the DAG already has it, and hence, can delete or flip it, or does not have it, and hence can add it.
  - So the number of operators in each step is  $O(n^2)$ .
- So a total of  $O(Kn^2)$  operations are applied.
- Now for each operator, two tasks are performed:
  - Checking the acyclicity of the resulting network. If we limit the number of parents to  $d$ , the total number of edges is  $nd$ . The complexity of acyclicity check can be shown to be  $O(nd)$ .
  - Comparing the score of the network with complexity order  $O(M)$ .

### 4.5.1 Score Function Decomposability

- What is the complexity of comparing the score of a DAG with the score of its neighbors. In general, we may need to compute separately the scores of the two DAGs and then compare them. However, the difference between the two DAGS is just a single edge.

#### Score Decomposability

- The score function  $S$  is decomposable if  $S(G|D)$  can be decomposed into the family scores of all variable in  $G$ .

$$S(G|D) = \sum_i \text{FamScore}(X_i, P_{a_{X_i}}|D) \quad (4.20)$$

where the family score of  $X$  is some function of  $X$ , its parents  $P_{a_X}$ , and the data  $D$ .

- The family score evaluates the fit of  $P_{a_X}$  as the parents of  $X$ .
- The likelihood score is decomposable with

$$\text{FamScore}_L(X_i|P_{a_{X_i}}|D) = N[I_{\hat{P}}(X_i, P_{a_{X_i}}) - H_{\hat{P}}(X_i)] \quad (4.21)$$

#### Benefits of Score Decomposability

- Let  $\Delta S(G; o)$  be the change that applying operator  $o$  on  $G$  create in the score:

$$\Delta S(G; o) = \text{score}(o(G)|D) - \text{score}(G|D) \quad (4.22)$$

- For the addition of edge  $X \rightarrow Y$ , we have

$$\Delta S(G; o) = \text{FamScore}(Y, P_{a_Y} \cup X|D) - \text{FamScore}(Y, P_{a_Y}|D) \quad (4.23)$$

- For the deletion of edge  $X \rightarrow Y$ , we have

$$\Delta S(G; o) = \text{FamScore}(Y, P_{a_Y} - X|D) - \text{FamScore}(Y, P_{a_Y}|D) \quad (4.24)$$

- For the reversal of edge  $X \rightarrow Y$ , we have

$$\begin{aligned} \Delta S(G; o) = & \text{FamScore}(X, P_{a_X} \cup Y|D) - \text{FamScore}(X, P_{a_X}|D) \\ & + \text{FamScore}(Y, P_{a_Y} - X|D) - \text{FamScore}(Y, P_{a_Y}|D) \end{aligned} \quad (4.25)$$

- Using the decomposability property of the score function, addition or deletion leads to changing only one local score term. Reversal leads to two score term changes.
- Thus, they result in at most two local changes in the score.
- So,  $O(M)$  is the same as the complexity order of computing a family score.

## 4.6 Summary: What is the Goal

- The goal is to systematically find a class PDAG P-map from data.
- One approach is constraint based

- One approach is score based. We need to search and find the bayesian network structure with the highest score.
- What are the scores? the likelihood score, bayesian score, and BIC.
- Given any of these scores, how can we perform a systematic search? Using search algorithms such as hill climbing and tabu search.