AMARTYA DUTTA

# Advanced Machine Learning

## HW : 2

### Problem : 1

Q1) The downside of Information Gain is that it favours the predictor variables with a large number of values. Thus, if the data is split using highly branching predictors, then the data might get split into subsets with low entropy. Thus, this would lead to possible overfitting or the number of nodes in the tree might may increase by a lot.

This is why Information Gain Ratio introduces a normalizing term, known as Split Info to reduce the bias.

$$\text{Split-Info}(F) = \sum \frac{|D_j|}{|D|} \log_2 \left( \frac{|D_j|}{|D|} \right)$$

$$\text{Information Gain Ratio} = \frac{\text{Information Gain}}{\text{Split Info}}$$

Finally, whichever predictor variable has the highest Gain Ratio is chosen as the predictor for splitting.

Q2) $\text{Entropy(Decision)} = -\sum p(I) \log_2 p(I)$

[where $I \in$ output classes]

$$= -p(\text{Yes}) \log_2 P(\text{Yes}) - p(\text{No}) \log_2 P(\text{No})$$

$$= -\left[\frac{9}{14}\log_2\left(\frac{9}{14}\right) + \frac{5}{14}\log_2\left(\frac{5}{14}\right)\right]$$

$$= 0.94$$

$$\text{Gain}(F) = \text{Entropy}(\text{Decision}) - \sum p(\text{Decision} \mid F) \times \text{Entropy}(\text{Decision} \mid F)$$

$$\text{Gain Ratio}(F) = \frac{\text{Gain}(F)}{\text{Split Info}(F)}$$

$$\text{Split Info}(F) = -\sum \frac{|D_j|}{|D|} \cdot \log_2\left(\frac{|D_j|}{|D|}\right)$$

Homework

$$\text{Gain}(D, \text{Homework}) = \text{Entropy}(\text{Decision}) - [p(\text{Decision} \mid \text{Homework} = \text{Much}) \times \text{Entropy}(\text{Decision} \mid \text{Homework} = \text{Much}) + p(\text{Decision} \mid \text{Homework} = \text{Normal}) \times \text{Entropy}(\text{Decision} \mid \text{Homework} = \text{Normal}) + p(\text{Decision} \mid \text{Homework} = \text{None}) \times \text{Entropy}(\text{Decision} \mid \text{Homework} = \text{None})]$$

→ Entropy (Decision | Homework = Much)

$$= -\left[\frac{2}{5}\log_2\left(\frac{2}{5}\right) + \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right]$$

$$= -\left[0.53 + 0.44\right]$$

$$= 0.97$$

→ Entropy (Decision | Homework = Normal)

$$= -\left[\frac{4}{4}\log_2\left(\frac{4}{4}\right) + 0\right] = 0$$

→ Entropy (Decision | Homework = None)

$$= -\left[\frac{3}{5}\log_2\left(\frac{3}{5}\right) + \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right] = 0.97$$

→ Gain (Decision, Homework) $= 0.94 - \left[\frac{5}{14} \times 0.97 + \frac{5}{14} \times 0.97 + \frac{4}{14} \times 0\right]$

$$= 0.94 - \left(\frac{5}{7} \times 0.97\right)$$

$$= 0.246$$

→ Split Info (Decision, Homework) $= -\left[\frac{5}{14}\log_2\left(\frac{5}{14}\right) + \frac{4}{14}\log_2\left(\frac{4}{14}\right) + \frac{5}{14}\log_2\left(\frac{5}{14}\right)\right] = 1.577$

Gain Ratio (Homework) = $\frac{0.246}{1.577}$ = **0.156**

Traffic

Gain (Decision, Traffic) = Entropy (Decision) − [p(Decision | Traffic = Busy) × Entropy (Decision | Traffic = Busy) + p(Decision | Traffic = Ok) × Entropy (Decision | Traffic = Ok) + p(Decision | Traffic = chill) × Entropy (Decision | Traffic = chill)]

→ Entropy (Decision | Traffic = Busy)

$= -\left[\frac{2}{4}\log_2\left(\frac{2}{4}\right) + \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right] = 1$

→ Entropy (Decision | Traffic = Ok) $= -\left[\frac{4}{6}\log_2\left(\frac{4}{6}\right) + \frac{2}{6}\log_2\left(\frac{2}{6}\right)\right] = 0.92$

→ Entropy (Decision | Traffic = chill) $= -\left[\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right]$

$= 0.81$

→ Gain (Decision, Traffic) = $0.94 - \left[\frac{4}{14} \times 1 + \frac{6}{14} \times 0.92 + \frac{4}{14} \times 0.811\right]$

$= 0.029$

→ Split Info (Decision, Traffic) $= -\left[\frac{4}{14}\log_2\left(\frac{4}{14}\right) + \frac{6}{14}\log_2\left(\frac{6}{14}\right) + \frac{4}{14}\log_2\left(\frac{4}{14}\right)\right]$

$= 1.56$

* Gain Ratio (Traffic) $= \frac{0.029}{1.56} = \boxed{0.0186}$

Hunger

Gain (Decision | Hunger) = Entropy (Decision) $- [p(\text{Decision} \mid \text{Hunger} = \text{A little}) \times \text{Entropy}(\text{Decision} \mid \text{Hunger} = \text{A little}) + p(\text{Decision} \mid \text{Hunger} = \text{A lot}) \times \text{Entropy}(\text{Decision} \mid \text{Hunger} = \text{A lot})]$

→ Entropy (Decision | Hunger = A little)

$= -\left[\frac{3}{7}\log_2\left(\frac{3}{7}\right) + \frac{4}{7}\log_2\left(\frac{4}{7}\right)\right]$

$= 0.985$

→ Entropy (Decision | Hunger = A lot)

$= -\left[\frac{6}{7}\log_2\left(\frac{6}{7}\right) + \frac{1}{7}\log_2\left(\frac{1}{7}\right)\right]$

$= 0.59$

→ Gain (Decision, Hunger) $= 0.94 - \left[\frac{7}{14} \times 0.98 + \frac{7}{14} \times 0.59\right]$

$= 0.15$

→ Split Info (Decision, Hunger) $= -\left[\frac{7}{14}\log_2\left(\frac{7}{14}\right) + \frac{7}{14}\log_2\left(\frac{7}{14}\right)\right]$

$= 1$

** Gain Ratio (Hunger) $= \boxed{0.15}$

Lauren

Gain (Decision, Lauren) = Entropy (Decision)

$- \left[P(\text{Decision} \mid \text{Lauren} = \text{Available}) \times \text{Entropy}(\text{Decision} \mid \text{Lauren} = \text{Available}) + p(\text{Decision} \mid \text{Lauren} = \text{Not Available}) \times \text{Entropy}(\text{Decision} \mid \text{Lauren} = \text{Not Available})\right]$

→ Entropy (Decision | Lawyer = Available)

$= -\left[\frac{6}{8}\log_2\left(\frac{6}{8}\right) + \frac{2}{8}\log_2\left(\frac{2}{8}\right)\right]$

$= 0.81$

→ Entropy (Decision | Lawyer = Not Available)

$= -\left[\frac{3}{6}\log_2\left(\frac{3}{6}\right) + \frac{3}{6}\log_2\left(\frac{3}{6}\right)\right]$

$= 1$

→ Gain (Decision, Lawyer) $= 0.94 - \left[\frac{8}{14} \times 0.8 + \frac{6}{14} \times 1\right]$

$= 0.048$

→ SplitInfo (Decision, Lawyer) $= -\left[\frac{8}{14}\log_2\left(\frac{8}{14}\right) + \frac{6}{14}\log_2\left(\frac{6}{14}\right)\right] = 0.98$

→ Gain Ratio (Lawyer) $= \frac{0.048}{0.98} = \boxed{0.048}$

As we can see that Homework has highest gain ratio, so homework will be the first split in the decision tree

Q3) From previous question, root node will be Homework; so the decision tree is:

(Homework = Much)

| Homework | Traffic | Hunger | Lauren | Go Out? |
|---|---|---|---|---|
| Much | Busy | A little | Available | No |
| Much | Busy | A little | Not Available | No |
| Much | OK | A little | Available | No |
| Much | Chill | A lot | Available | Yes |
| Much | OK | A lot | Not Available | Yes |

Thus, going out has a true correlation with Hunger, if Homework is much

(Homework = Normal)

| Homework | Traffic | Hunger | Lauren | Go Out? |
|---|---|---|---|---|
| Normal | Busy | A little | Available | Yes |
| Normal | Chill | A lot | Not available | Yes |
| Normal | OK | A little | Not available | Yes |
| Normal | Busy | A lot | Available | Yes |

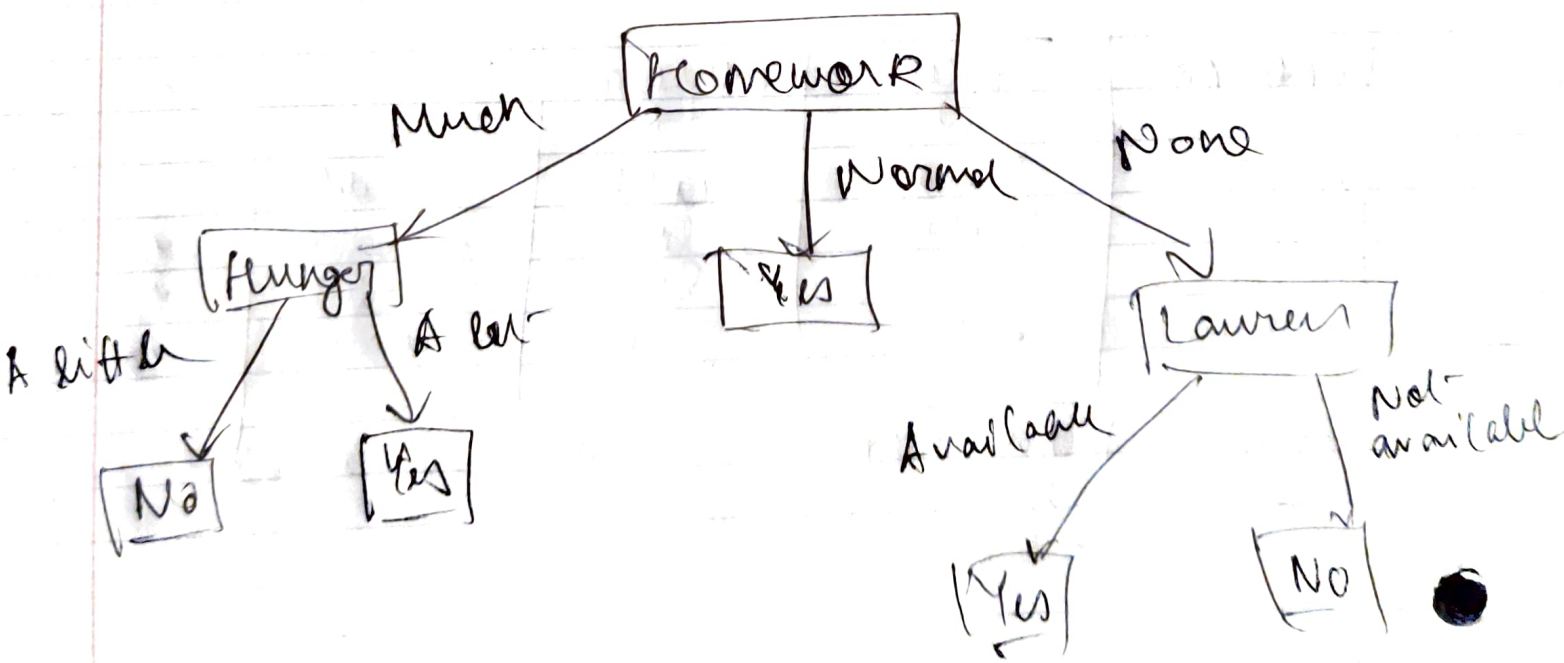
Thus, if Homework is Normal, then he will always go out.

(Homework = None)

| Homework | Traffic | Hunger | Lauren | Go Out? |
|---|---|---|---|---|
| None | OK | A little | Available | Yes |
| None | Chill | A lot | Available | Yes |
| None | Chill | A lot | Not available | No |
| None | Ok | A lot | Available | Yes |
| None | Ok | A little | Not available | No |

From the table ~~we will go out~~ going out is directly correlated to Lauren if there is no Homework.

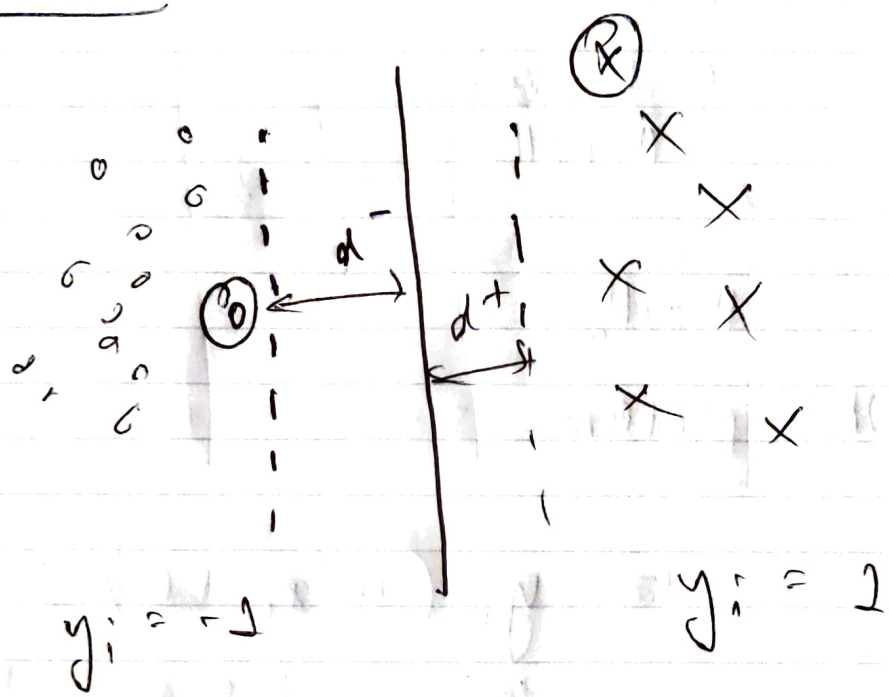
Thus, the new decision tree is:



Q4) Since amount of homework is normal I will always go out as per the decision tree.

## Problem 2

Q1)




⇒ Linearly seperable data

For class $y_i = 1$

$$w^T x_i + b \geq 1$$

$$w^T x_j + b = 1 \Rightarrow b = 1 - w^T x_j$$

Similarly, for class $y_i = -1$

$$w^T x_j + b \leq -1$$

$$w^T x_k + b = -1 \Rightarrow b = -1 - w^T x_k$$

$\therefore d^+ - d^-$

$= \frac{w^T x_i}{\|w\|_2} - \frac{w^T x_k}{\|w\|_2}$

$= \frac{1-b}{\|w\|_2} - \frac{(-1-b)}{\|w\|_2}$

$\Rightarrow d^+ - d^- = \frac{2}{\|w\|_2}$

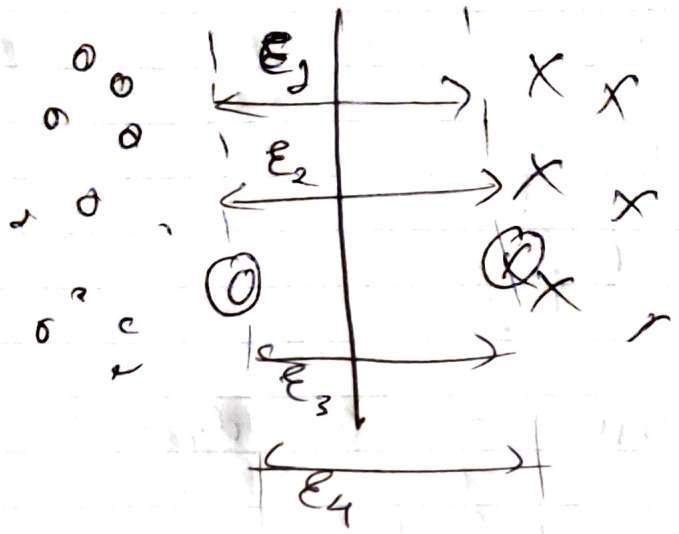
$\therefore$ The SVM problem is $\rightarrow$

$\max_{w,b} \frac{2}{\|w\|_2}$ s.t $y_i(w^T x_i + b) \geq 1$
for $i \in \{1, \ldots N\}$
no of data points

or

$\min_{w,b} \frac{1}{2}\|w\|_2^2$ s.t $y_i(w^T x_i + b) \geq 1$
for $i \in \{1, \ldots N\}$

Q2)




If the data is not linearly separable then we have to further deal with slack variables. The slack variables are used to minimize the misclassifications along with the objective function defined in the above problem

$$\frac{1}{2}\|w\|_2^2$$

In the case data points are correctly classified,

$$y_i(w^T x_i + b) \geq 1$$

else, $y_i(w^T x_i + b) < 1$

The slack variables that we will use can be defined as:

$\xi_1, \xi_2 \ldots \xi_N$ for misclassifiers

(for $i \in 1, \ldots N$)

∴ Now dyc $f^n$ is :

$$\frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \varepsilon_i$$

∴ The SVM problem can be defined as :

$$\min_{w, b, \varepsilon_i} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \varepsilon_i$$

$$\text{s.t } y_i(w^T x_i + b) \geq 1 - \varepsilon_i$$

$$\text{for } i \in \{1, \ldots N\}$$

$$\varepsilon_i \geq 0 \text{ for } i \in \{1, \ldots N\}$$

Q3) SVM doesn't normally help with multiclass classification. However, it can be broken down into smaller binary classification subproblems. There are several approaches like One vs One, One vs All and Directed Acyclic Graph.

→ In One vs One we breakdown the multiclass classification problem into various binary classification. Finally, majority voting is done.

→ In the One vs All eg, we train N SVMs

SVM (1) → for class 1

⋮

SVM(N) → for class N

Now, in order to predict the output of new input, predict the output using all the SVMs and then just identify which model pulls the prediction farthest into the true region of the data.

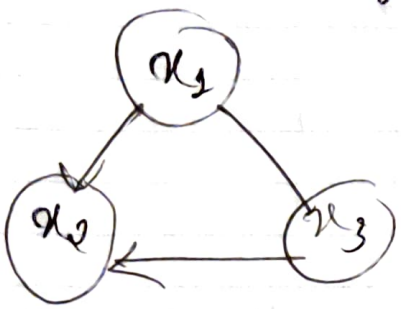
→ In Directed Acyclic graphs, the classes are first grouped based on some grouping and then train the SVMs. Thus, at the end we might need to train less number of SVMs and this approach reduces the diversity from the majority class.

Problem 3:

Q1)

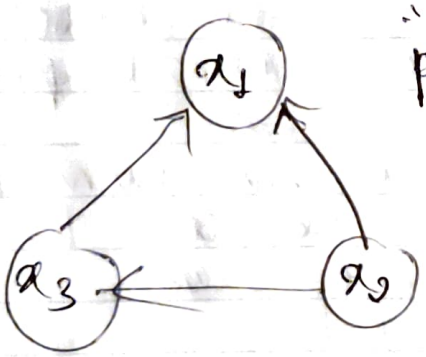
(1) We are given N random variables $x_1, x_2 \ldots x_N$. And thus, the aim is to create a Bayesian Network showing the joint dist$^n$ of all N variables.

Let us consider, joint dsn of $x_1, x_2, x_3$




$\therefore p(x_1, x_2, x_3) = p(x_1) \cdot p(x_2|x_1)\, p(x_3|x_1, x_2)$

Similarly, if the ordering of the graph is changed:




$\therefore p(x_1, x_2, x_3)$

$= p(x_3) \cdot p(x_3|x_2) \cdot p(x_1|x_3, x_2)$

Thus, if we listing out all possibilities, thus for 3 random variables there are $3! = 6$ distinct graphs.

So for N random variables, N! graphs

② For a single discrete variable $x$ having $M$ possible states, the probability distribution is

$$P(x|\sigma) = \prod_{m=1}^{M} \mu_m^{x_m}$$

Similarly, for discrete random variables $x_1$, $x_2$ we have two states: $S_1$ & $S_2$

$$\therefore P(x_1, x_2|\sigma) = \prod_{a=1}^{S_1} \prod_{b=1}^{S_2} \mu_{ab}^{x_{1a} x_{2b}}$$

& we can see that total parameters are $\rightarrow (S_1 \times S_2 - 1)$

Thus, after generalising this we can say that if we have $N$ random variables $x_1, x_2 \ldots x_N$ & they have $N$ states $S_1, S_2 \ldots S_N$ then total no. of parameters in the joint probability dist$^n$ will be:

$$[(S_1 \times S_2 \ldots \times S_N) - 1]$$

This shows that we will eventually end up with a graph that grows exponentially and so the task to calculate joint probability in the case will become very complex.

Now if we consider all our random variables to be conditionally independent then

for two random variables $x_1, x_2$

joint probability $p(x_1, x_2 | \theta) \rightarrow$

$$p(x_1, x_2 | \theta) = \prod_{a=1}^{S_1} \theta_{1a}^{x_{1a}} \prod_{b=1}^{S_2} \theta_{2b}^{x_{2b}}$$

& no. of parameters in this joint probability will be $(S_1 + S_2 - 1)$

∴ N random variables with N states will have $\Rightarrow (S_1 + S_2 + \cdots S_{N-1})$ param

& thus, the graph will grow linearly instead of exponentially.

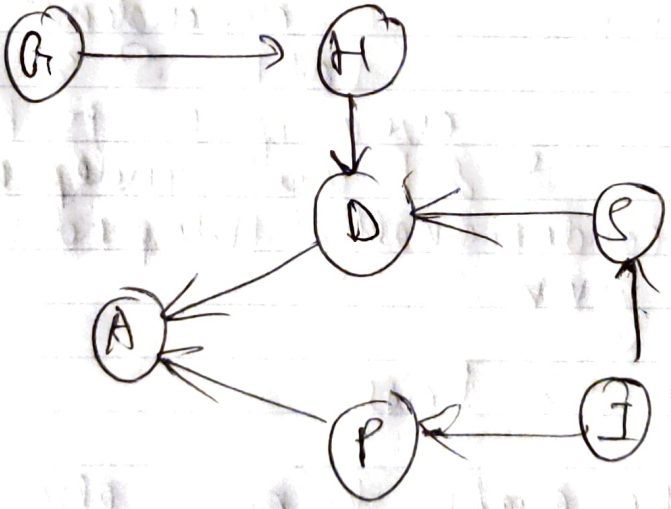
Hence if we maintain conditional independence of all N random variables, then it will be less complex to calculate joint probability $p(x_1, x_2 \ldots x_N)$ as the graph will be linear & not exponential.

Q2)

(1)

G ⇒ Going out right now
H ⇒ contracting Hokie plague
D ⇒ Drowsy
S ⇒ Lack of Sleep
I ⇒ Insomnia
P ⇒ Pay Attention
A ⇒ Unable to finish the assg

Bayesian Network




$$P(G,H,D,S,I,P,A) = P(G) \cdot P(H|G) \cdot P(D|H,S) \cdot P(S|I) \cdot P(I) \cdot P(P|I) \cdot P(A|P,D)$$

Nodes in the S's Markov Blanket ⇒

I, D & H as if we know the values of these nodes, then we can determine the value of S.

(2) S ⫫ P | J → True

Simplifying the graph to get above nodes, we get ⇒

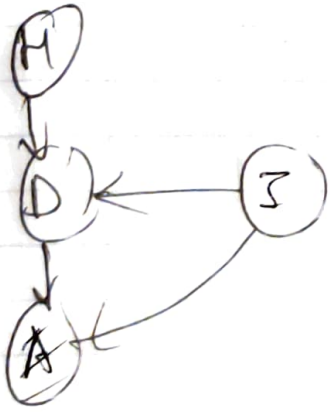

Sine, I is dependent on both S & P even if I is known S & P are independent

~~So even~~

∴ S is conditionally independent of P given I

(3) H ⫫ I | A → False

Simplifying the graph to get above nodes we get ⇒

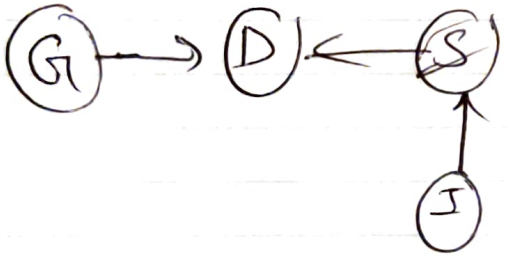



Sine, A is dependent on D & I & D is dependent on both I & H.

∴ H is also dependent on I when A is known

∴ H is conditionally dependent on I given A.

(4) G ⫫ I | S → True

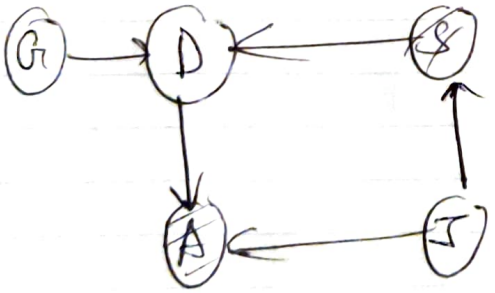



Since, D is dependent on both G & S and S is dependent on I

Hence, G is independent of I even when S is given.

∴ G is conditionally independent on I given S.

(5) G ⫫ I | S, A → False




Since, A is dependent on I & D and D is dependent on both S & G and S is dependent on I

Hence, G is dependent on I even when S & A are known.

∴ G is dependent on I given S and A.