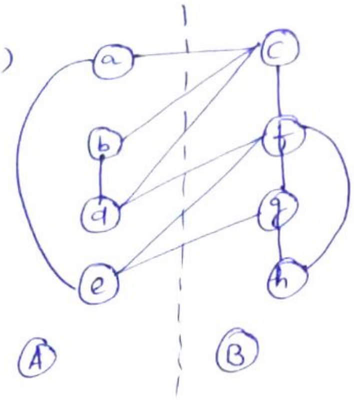
P V Sri Ram
B Tech (4th 1st sem)
1801 CS 37

1 A)



$\boxed{\text{Cut Size} = 7}$ (0th)

→ Partition A

Node	k^{other}	k^{same}	$\Delta = (k^{other} - k^{same})$
a	1	1	0
b	1	1	0
d	2	1	1
e	3	1	2

→ Partition B

Node	k^{other}	k^{same}	$\Delta = (k^{other} - k^{same})$
c	4	1	3
f	2	3	-1
g	1	2	-1
h	0	2	-2

Computing gains $G_{i,j} = k_i^{other} - k_i^{same}$

$$+ k_j^{other} - k_j^{same}$$

$$- 2A_{ij}$$

$$= \boxed{\Delta_i + \Delta_j - 2A_{ij}}$$

$G_{ac} = 0+3-2(1) = 1$

$G_{af} = 0+(-1)-2(0) = -1$

$G_{ag} = 0+(-1)-2(0) = -1$

$G_{ah} = 0+(-2)-2(0) = -2$

$G_{bc} = 0+3-2(1) = 1$

$G_{bf} = 0+(-1)-2(0) = -1$

$G_{bg} = 0+(-1)-2(0) = -1$

$G_{bh} = 0+(-2)-2(0) = -2$

$G_{dc} = 1+3-2(1) = 2$

$G_{df} = 1+(-1)-2(1) = -2$

$G_{dg} = 1+(-1)-2(0) = 0$

$G_{dh} = 1+(-2)-2(0) = -1$

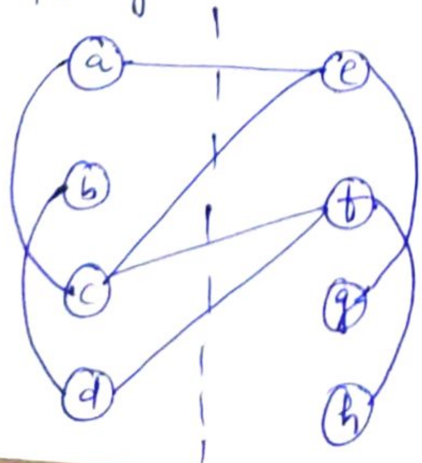
$G_{ec} = 2+3-2(1) = 3$ ← for Best

$G_{ef} = 2+(-1)-2(1) = -1$

$G_{eg} = 2+(-1)-2(1) = -1$

$G_{eh} = 2+(-2)-2(0) = 0$

→ Swapping e & c



1st

$\boxed{\text{Cut size} = 4}$

e & c are now fixed.

→ Partition A (not fixed)

Node	k^{other}	k^{same}	Δ
a	1	1	0
b	0	2	-2
d	1	2	-1

→ Partition B (Not fixed)

Node	k^{other}	k^{same}	Δ
f	2	3	-1
g	0	3	-3
h	0	2	-2

$G_{af} = 0 + (-1) - 2(0) = -1$ ← best

$G_{ag} = 0 + (-3) - 2(0) = -3$

$G_{ah} = 0 + (-2) - 2(0) = -2$

$G_{bf} = -2 + (-1) - 2(0) = -3$

$G_{bg} = -2 + (-3) - 2(0) = -5$

$G_{bh} = -2 + (-2) - 2(0) = -4$

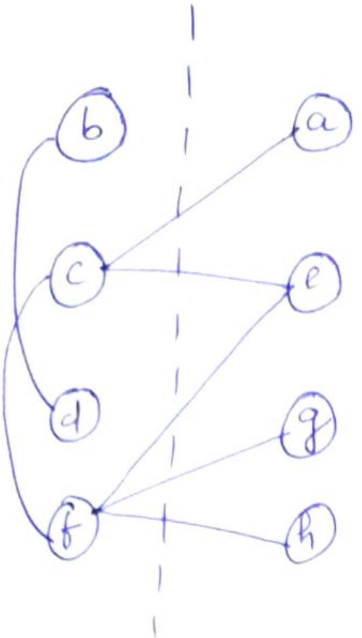
$G_{df} = -1 + (-1) - 2(1) = -4$

$G_{dg} = -1 + (-3) - 2(0) = -4$

$G_{dh} = -1 + (-2) - 2(0) = -3$

Swapping a & f and fixing them

Swapping (b & h) and fixing them.



2nd.

$\boxed{\text{Cut Size} = 5}$

→ Partition A (Not fixed)

Node	k^{other}	k^{same}	Δ
b	0	2	−2
d	0	3	−3

→ Partition B (Not fixed)

Node	k^{other}	k^{same}	Δ
g	1	2	−1
h	1	1	0

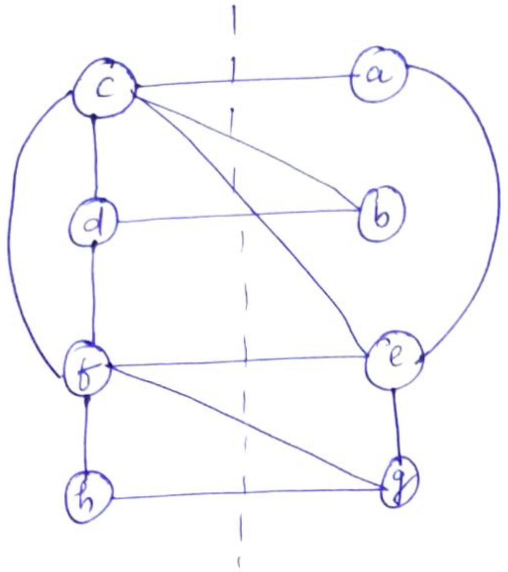
$G_{bg} = -2 + (-1) - 2(0) = -3$

$G_{bh} = -2 + 0 - 2(0) = -2$ ← best

$G_{dg} = -3 + -1 - 2(0) = -4$

$G_{dh} = -3 + 0 - 2(0) = -3$

Swapping (b & h) and fixing them.



3^{rd}

Cut size = 7

Now d, g are the only vertices left to swap, swapping them will make the partitioning cut size same as initial one (i e Cut size = 7) because only the name of partition is different.

4^{th}

Cut size = 7

Final Ans:-

(i, j) Edge	Δ (Total change in cut size)
(e, c)	−3
(a, f)	1
(b, h)	2
(d, g)	0

1 (b) Derive the graph Laplacian matrix L for the given graph and show $R = \frac{1}{4} \vec{s}^T L \vec{s}$

$R = \frac{1}{2} \sum_{i,j} A_{ij}$, where i and j belong to different groups

Let's define:-

$$S_i \begin{cases} +1, & \text{if vertex } i \text{ belongs to group 1} \\ -1, & \text{if vertex } i \text{ belongs to group 2} \end{cases}$$

Then :-

$$\frac{1}{2}(1 - S_i S_j) \begin{cases} 1, & \text{if } i \,\&\, j \text{ are in different groups} \\ 0, & \text{if } i \,\&\, j \text{ are same group} \end{cases}$$

$$R = \frac{1}{4} \sum_{ij} A_{ij} (1 - S_i S_j)$$

$$\sum_{ij} A_{ij} = \sum_i k_i = \sum_i k_i S_i^2 = \sum_{ij} k_i \delta_{ij} S_i S_j$$

$$R = \frac{1}{4} \sum_{ij} (k_i \delta_{ij} - A_{ij}) S_i S_j$$

$$= \frac{1}{4} \sum_{ij} L_{ij} S_i S_j \Rightarrow R = \frac{1}{4} \vec{S}^T L \vec{S}$$

• L is Laplacian Matrix

Since S_i can only take integral values $[-1,1]$, we treat this as an integer optimisation problem. This can be solved using relaxation method.

Our Objective function is

$$\boxed{R = \frac{1}{4}\vec{S}^T L \vec{S}} \quad \text{--- (1)}$$

We have to minimise R

Now we need to specify the constraints

* In the original integer problem the vectors would point to one of the 2^n corners of an n-dimensional hyper cube centred at the origins

The length of $\vec{S}$ is $\sqrt{n}$

We relax the above constraint by allowing the vector is to point to any direction as long as its length is still $\sqrt{n}$.

$$\boxed{\sum_i S_i^2 = n} \quad \text{--- (2)}$$

* A second constraint in the integer optimization problem is the fact that the number of elements in vectors that equal $+1$ or -1 needs to be equal to the desired sizes of the two group

$$\boxed{\sum_i S_i = n_1 - n_2 \Rightarrow \vec{1}^T \vec{S} = n_1 - n_2} \quad \text{--- (3)}$$

8

From 4 we know

$$L_{ij} = (k_1 \delta_{ij} - A_{ij})$$

K =

	a	b	c	d	e	f	g	h
a	2	0	−1	0	−1	0	0	0
b	0	2	−1	−1	0	0	0	0
c	−1	−1	5	−1	−1	−1	0	0
d	0	−1	−1	3	0	−1	0	0
e	−1	0	−1	0	4	−1	−1	0
f	0	0	−1	−1	−1	5	−1	−1
g	0	0	0	0	−1	−1	3	−1
h	0	0	0	0	0	−1	−1	2

2 ## Clustering Coefficient:-

clustering coefficient is defined as the probability that two vertices with a common neighbour are connected themselves.

Local Actual clustering of node i

$$= \frac{(\text{Actual number of links between neighbours of } i)}{(\text{Maximum number of possible links between neighbours of } i)}$$

(a) Given, a $G(n, P)$ model where

P = Probability that any 2 nodes in a network are connected

Expected number of links b/w k_i neighbours of node i,

$$\langle k_i \rangle = p \frac{k_i(k_i - 1)}{2} \quad \text{——} \quad (1)$$

where k_i = degree of node i

maximum number of possible links $= \frac{k_i(k_i-1)}{2}$

Local clustering

Coefficient of node i $= C_i = \dfrac{2 \langle k_i \rangle}{k_i (k_i - 1)} = p$

mean degree is given by

$$\langle k \rangle = \sum_{m=0}^{\binom{n}{2}} \frac{2m}{n} P(m) = \frac{2}{n} \binom{n}{2} p = (n-1)p$$

$$\Rightarrow p = \frac{\langle k \rangle}{n-1}$$

$\therefore$ Clustering Coefficient $= C_i = \dfrac{2 \langle L_i \rangle}{k_i (k_i - 1)} = p = \dfrac{\langle k \rangle}{N-1}$

where $\langle k \rangle$ = mean degree,

N = number of vertices.

11 6

(b) Given that,

$N \to$ large

Mean degree, $\langle k \rangle$ is constant

from (a) we know,

clustering coefficient $= P = \dfrac{\langle k \rangle}{N-1}$

as $N \to \infty$, $\langle k \rangle \to$ const $\Rightarrow P \to 0$

$\therefore$ For large n, $\langle k \rangle$ is constant then the "clustering coefficient goes to 0"

2 (c)

Let us take a random graph with 'c' average degree.

For a given node:-

c^0 nodes are at a distance of 0 from it ($d=0$)

c^1 nodes are at a distance of 1 from it ($d=1$)

c^2 nodes are at a distance of 2 from it ($d=2$)

c^3 nodes are at a distance of 3 from it ($d=3$)

⋮

c^λ nodes are at a distance of λ from it ($d=\lambda$)

No. of nodes upto a distance λ from the given node are :-

$$N(\lambda) = 1 + c + c^2 + \dots c^\lambda = \frac{c^{\lambda+1} - 1}{c - 1}$$

If we take $\lambda = d_{max}$ [Diameter of graph)

then $N(d_{max}) = N$

This is because the furthest node from the given node lies at a distance of d_{max} [definition of diameter of graph] and all other nodes would be at a distance equal or less than d_{max}

$$\therefore N(d_{max}) = N.$$

$$N = \frac{c^{d_{max}+1} - 1}{c - 1}$$

$$\Rightarrow \quad c^{d_{max}+1} = 1 + N(c-1)$$

$$\simeq Nc \quad \text{as } N \gg 1$$

$$\Rightarrow \quad \left(c^{d_{max}}\right) c \simeq Nc$$

$$\Rightarrow \quad d_{max} \simeq \frac{\ln N}{\ln c}$$

(9)

Assume P_k = Probability that a node is connected to exactly k other vertices

So, Total probability of being connected to set of k vertex among $n-1$ possible neighbours

$$P_k = \binom{n-1}{k} p^k (1-p)^{n-1-k} \quad \text{——— (1)}$$

if $n \to$ large, then network has constant average degree c

$$c = (n-1)p \Rightarrow p = \frac{c}{n-1}$$

As $n \to \infty$

$$\binom{n-1}{k} = \frac{(n-1)!}{k!\,(n-1-k)!}$$

$$= \frac{(n-1)(n-2)\cdots(n-k)\cancel{(n-1-k)!}}{k!\,\cancel{(n-1-k)!}}$$

$$\therefore \lim_{n\to\infty} \binom{n-1}{k} \cong \frac{n^k}{k!}$$

Now,

$$\lim_{n\to\infty} (1-p)^{(n-1-k)} = \lim_{n\to\infty} \left(1 - \frac{c}{n-1}\right)^{n-1-k}$$

(Substituting (1))

$$\Rightarrow \lim_{n\to\infty} \left(1 - \frac{c}{n}\right)^{n} = e^{-c} \quad \text{——— (3)}$$

(10) (15)

From (1),

$$P_k = \binom{n-1}{k} p^k (1-p)^{(n-1-k)}$$

as $n \to \infty$, $P_k = \lim_{n\to\infty} \frac{n^k}{k!} \left(\frac{c}{n-1}\right)^k e^{-c}$

$$= \lim_{n\to\infty} \left(\frac{n}{n-1}\right) \frac{c^k e^{-c}}{k!}$$

$$\boxed{P_k = \frac{c^k e^{-c}}{k!}} \longrightarrow \text{Poission Distribution}$$

$\therefore$ Degree distribution of nodes follows a poisson distribution

16

2 e) A network component whose size grow in proportion of to 'n' is a giant component

→ let its be the fraction of nodes that do not belong to the giant component

⇒ if there is no giant component $u = 1$

⇒ if there is a giant component $u < 1$

→ In order for a node i not to be a part of the giant component

a) A = i should not be connected to any node
$$P(A) = 1 - P$$

b) B = i is connected to node, but j is not a part of giant component
$$P(B) = pu$$

→ The total probability of i not being connected to giant component are vertex j is $1 - p + pu$

→ Extending j over all nodes

$$u = (1 - p + up)^{n-1} = \left[1 - \frac{c}{n-1} + \frac{cu}{n-1}\right]^{n-1}$$

$$\Rightarrow u = \left[1 - \frac{c}{n-1}(1-u)\right]$$

(17)

Taking log on both sides

$$\Rightarrow \log u = (n-1) \log \left(1 - \frac{c}{n-1}(1-u)\right)$$

[Assuming n is large]

$$\Rightarrow \log u \approx -(n-1)\left[\frac{c}{n-1}(1-u)\right]$$

[$\log(1-x) \approx -x$ for $x << 1$]

$$\Rightarrow \log u \propto -c(1-u)$$

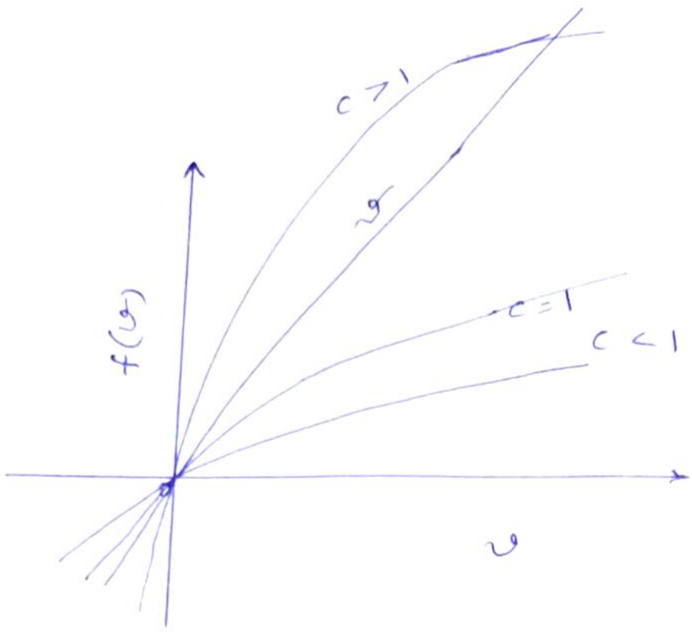
$$\Rightarrow u \simeq e^{-c(1-u)}$$

taking $V = 1-u$

$$\Rightarrow V = 1 - e^{-Vc}$$

This equation has a solution at $v=0$ which is trivial then $u=1$, which means our graph has no connected nodes

plotting V and $1-e^{-V}$ graph as a function of V



So, $c \leq 1$ is guaranteed to have no solution is seen from the graph and we can see for $c > 1$, another solution (other than $\vartheta = 0$) exists

Intution for $c > 1$ having a solution

→ C is the mean degree and $c > 1$, implies that on an average every node has more than 1 neighbour

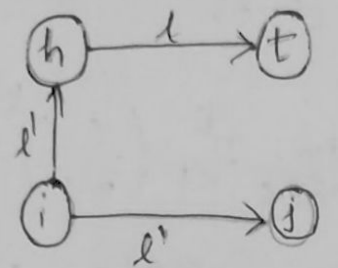
→ Taking a node it will have 'c' neighbours. Its neighbours will each be having 'c' neighbours making it c^2 neighbours

→ If we continue like this for k steps then c^k nodes should be reached at a distance of k from the initial vertex.

→ As $c > 1$, the number would keep on increasing, and so most of the nodes would be connected into a giant component.

→ So the required condition is $c > 1$

3 a) For proving that the loss function L_{single} yields useless embeddings, we consider the following graph:-



In the above graph, we consider four nodes h, t, i and j and consider two different kinds of relations, l and l'

Considering that we need to optimize the loss function:-

$$L_{single} = \sum_{(h,l,t) \in S} d\,[h+l, t]$$

where $d(.)$ represents the Euclidean distance between $h+l$ and t, we can see that, to optimize the function $h+l$ must be very close to t in the embedding space. For this we get the following relation:-

$$h + l - t = \epsilon\, 1 \quad (or) \quad h + l \simeq t$$

[ϵ is a small valued scalar, 1 is the vector $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$]

Similarly, we obtain from the other relations the following set of equations :-

$h+e-t=t$ or $h+e=t$ — ①

$i+e'-h=t$ or $i+e'=h$ — ②

$i+e'-j=t$ or $i+e'=j$ — ③

Now, we take the embedding of h as $\begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix}$

i as $\begin{bmatrix} \frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix}$ and t as $\begin{pmatrix} -1 \\ 0 \end{pmatrix}$ [Considering all embeddings are of length 1]

From this we get $h+e=t$ $\begin{pmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{pmatrix} + \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$

$\Rightarrow l = \begin{pmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{pmatrix}$

and, $i+l'=h \Rightarrow \begin{pmatrix} \frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{pmatrix} + \begin{pmatrix} l_1' \\ l_2' \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{pmatrix}$

$\Rightarrow l' = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$

and $i+l'=j \Rightarrow \begin{pmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{pmatrix} + \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ -\sqrt{3}/2 \end{pmatrix}$

After obtaining l, l' and j, we make the following observations:-

(i) h and j are having similar embeddings (if we don't consider the effect of ϵ, then they will be exactly similar)

(ii) We are getting that $j+l=t$ (or $j+l-t=t$)

Now, we see that j is holding the same relation with t over l, as h is holding.

This means that, the loss function is not able to optimize the solution to the problem of finding good node embeddings. According to the passage, in ideal case, j should not hold the same relation with t over link l, as j and t are not corrected by the link 'l' at all.

Hence, we can see that the node embeddings learnt are not indicating whether two nodes are linked by a particular relation. Since the embeddings do not highlight the difference between nodes of the actual graph that are corrected by the relation (h over l), as compared to nodes not connected (j over l) with a node (t), they are useless.

Hence, the objective is simple will yield a useless embeddings.

14 21

3 (b)

Given, for a multi relational graph based learning, we are using TransE algorithm.

Consider Graph, $G = (E, S, L)$

where $E \to$ set of nodes

$S \to$ set of edges

$L \to$ set of relations

Required to learn embeddings $e \in E$, $l \in L$ in $\mathbb{R}^k$ space

Now consider the loss function,

$$\mathcal{L}_{no\text{-}margin} = \sum_{(h,l,t) \in S} \left(\sum_{(h',l,t')} \left[d(h+l,t) - d(h'+l,t') \right] \right)$$

where S' are corrupted edges.

RTP:- The above loss function will result in dis-functional embeddings

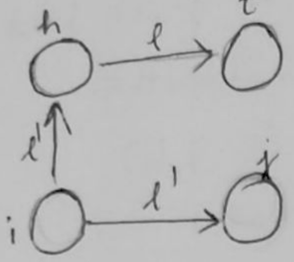
Now, the above loss function will converge in three scenarios.

Case ①: $d(h+l,t) = d(h'+l,t') \neq 0$

Case ②: $d(h+l,t) = d(h'+l,t') = 0$

Case ③: $d(h+l,t) \to 0$, $d(h'+l,t') >> 0$

(a) Consider the following graph



$E = \{t, h, i, j\}$

$S = \{(i,j), (i,h), (h,t)\}$

$L = \{l, l'\}$

Relations,

Case - 1	Case - 2	Case - 3
$d(h+l, t) = K$ $d(j+l, t) = K$ $K > 0$ Non-ideal Convergence Since (h,t) are Considered not Connected	$d(h+l, t) = 0$ $d(j+l, t) = 0$ Non-ideal Convergence Since (j,t) are Considered Connected	$d(h+l, t) = 0$ $d(j+l, t) > 0$ Ideal Convergence Since (h,t) are Considered Connected and (j,t) are considered not Connected.

(b) Example

Case (1)

$h = \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$

$i = \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$

$t = (+1, 0)$

$j = \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$

$l' = (-1, 0)$

Case (2)

$h = \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$

$i = \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$

$t = (-1, 0)$

$j = \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$

$l' = (-1, 0)$

c) Ideally, we need $d(h+l, t) = 0$; $d(j+l, t) > 0$
(connected) (disconnected)

However in case 1 $d(h+l, t) = d((-1,0), (+1,0))$
$= 2 \neq 0$

and in case 2 $d\,e(j+l, t) = d((-1,0), (-1,0)) = 0$

$\therefore$ The embeddings achieved through convergence of $L_{\text{no-margin}}$ are not functional.

This could be corrected by assigining a penalty margin 'γ' to loss.