QUIZ 10 SOLUTION

1) AGM - generative model $B(V, C, M, \{P_c\})$ —— [1 POINT]

V - no of nodes

C - no of communities

M - Memberships

P_c - Each community c has a single probability P_c

AGM has to find a model M given a graph $G(V,E)$ by assuming the variables M, C and P_c.

MLE - given data X.

Assumption - data generated by some model $f(\theta)$ → parameters

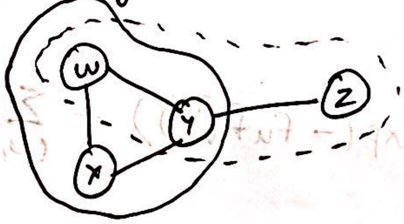
Our goal is to find $P_f(X|\theta)$ ie. probability that our model f (with θ parameters) generated the data.

Our goal is to find the $\arg_\theta \max P_f(X|\theta)$ —— [2 POINTS]

also, $\arg\max_{B(V,C,M,\{P_c\})} \prod_{u,v \in E} P(u,v) \prod_{u,v \notin E} (1 - P(u,v))$

We repeatedly perform MLE, to test against as many assumptions as possible to find the optimal value of $P_f(X|\theta)$ —— [1 POINT]

example:



$C = \{w, x, y\}$

$D = \{w, y, z\}$

P_c & P_D

→ Assumptions.

← temporary community memberships

The likelihood of this graph given the assumption

$L = P_{xy} P_{wu} P_{xv} P_{vz} (1 - P_{wz})(1 - P_{xz})$

$L = (P_c)^2 P_D (P_c + P_D - P_c P_D)(1 - P_D)(1 - \epsilon)$

$P_c = 1$, as large as possible maximises the above equation.

$\therefore P_D(1 - P_D) \Rightarrow P_D = 0.5$

We repeat this process for $\forall$ diff assumptions and find the model that gives the maximum L. —— [1 POINT]

2) **BigCLAM** -

→ avoids discrete memberships and considers the membership strengths.

F_{uA} → The membership strength of a node u to community A (>0)

if $F_{uA} = 0$ means no membership — [1 POINT]

→ It uses membership strength matrix F

$F = [\ ]$ nodes × communities

(F_{uA}) Strength of u's membership to A ↳ [1 POINT]

→ Each community A links nodes independently

We define Prob. of u and v in community A by

$$P_A(u,v) = 1 - \exp(-F_{uA} \cdot F_{vA})$$

example :-

$$\begin{array}{c} F_u \\ F_v \\ F_w \end{array} \begin{bmatrix} 0 & 1.2 & 0 & 0.2 \\ 0.5 & 0 & 0 & 0.8 \\ 0 & 1.8 & 1 & 0 \end{bmatrix}$$

then, $F_u \cdot F_v^T = 0.16$

$P(u,v) = 1 - \exp(-0.16)$
$= 0.14$

$P(u,w) = 0.88$

$P(v,w) = 0$

→ Find F that maximizes $l(F)$

$$l(F) = \sum_{(u,v)\in E} \log(1 - \exp(-F_u F_v^T)) - \sum_{(u,v)\notin E} F_u F_v^T$$

→ $$\nabla l(F_u) = \sum_{v \in N(u)} \frac{F_v \exp(-F_u F_v^T)}{1 - \exp(-F_u F_v^T)} - \sum_{v \notin N(u)} F_v$$ —[1 POINT]

↳ compute gradient of a single row F_u of F

→ Iterate over the rows of F — [1 POINT]

→ compute $\nabla l(F_u)$ of each row u

→ update $F_u \leftarrow F_u + \eta \nabla l(F_u)$

→ Project F_u back to non-ve vector

if $F_{uc} < 0$: $F_{uc} = 0$.

used for large scale Networks

→ Cache $\sum_v F_v$ so, computing $\sum_{v \in N(u)} F_u$ takes linear time

(i.e the degree of each node in a large Network is less than the no of nodes in the network) [1 POINT]