Clustering based on Random Graph model embedding vertex features.

1. Introduction / Abstract:

This paper is motivated by a problem of network clustering where each node in a network has a covariate or covariates. That is, given some adjacency matrix x ($n \times n$ $i,j \in \{1,2,\dots,n\}$) where $\text{diag}(x) = 0$ and x is symmetric and some node specific features as a set $y = \{y_1, y_2, \dots, y_n\}$ where each y_i is p-dimensional vectors of features, How can we use this information to find clusters?

They propose to use classical Stochastic Blockmodel (SBM) and extend it for this case. They call this new extension the Mix Net model.

2 A mixture of Network with covariates

The MixNet assumes that the covariates y and edges x are independent given the node clusters. Or in other words that the connecting profile of a given node and its features can be explained by its cluster.

To comment on this further, we introduce some notation.

- x is $n \times n$ matrix $i,j \in \{1,2,\dots,n\}$ nodes Thus $x_{ij} = 1$ or $x_{ij} = 0$.
  X is $n \times n$ random matrix
- z is $n \times Q$ matrix of $i \in \{1,2,\dots,n\}$ $q,l \in \{1,2,\dots,Q\}$ clusters.
  Thus $z_1 = (0,1,0,0)$ means there was $Q=4$ and node 1 is located in the 2nd class.
  Z is random variable
- y is $n \times p$ matrix of features $i \in (1,2,\dots,n\}$ $p \in \{1,\dots,P\}$.
  Y is random variable.

Distributions:

- $X_{ij} \mid Z_{iq}=1, Z_{jl}=1 \sim$ Bernoulli (π_{ql}) ← small letters.

$$f(x \mid z; \pi) = \prod_{i=1}^{n} \prod_{j \neq i} \prod_{q=1}^{Q} \prod_{l=1}^{Q} f(x_{ij} \mid z_{iq}=1, z_{jl}=1)^{z_{iq} z_{jl}} \quad (EQ.1)$$

- $z_i \sim$ Categorical (α), where α is a vector of length Q. (EQ.2)
- $(y_i \mid z_{iq}=1) \sim MVN(\mu_q, \Sigma_q)$ $f(y \mid z; \mu, \sigma^2) = \prod_{i=1}^{n} \prod_{q=1}^{Q} f(y_i \mid z_{iq})^{z_{iq}}$ (EQ 3)

for each group variance component

$\mu_q = (\mu_q^{(1)}, \dots, \mu_q^{(P)})$ and $\Sigma_q = \sigma_q^2 I$ ← $p \times p$ matrix.

The latter assumption $\Sigma_q = b^2 I$ means that all features within a cluster 2 have a common variance and moreover that this variance is constant within a clusters. Thus, it is important to standardise node features before the analysis. (3)

Model's likelihood.

$$\log f(x, y, z; \pi, M, \sigma^2, \alpha) = \log f(x, y \mid z; \pi, M, b^2)\, f(z; \alpha)$$
$$= \log f(x \mid z; \pi) + \log f(y \mid z; M, b^2) + \log f(z; \alpha) \quad \text{Eq (4)}.$$

In particular, they are interested in optimising marginal log-likelihood:

$$\log f(x, y; \pi, M, b^2, \alpha) = \log \sum_{z \in S} f(x, y, z; \pi, M, b^2, \alpha)$$

which is not tractable. So they propose to use variational approximation In general, variational approximation is understood in the following way:

$$\log f(x, y; \pi, M, b^2, \alpha) = \log \sum_{z \in S} f(x, y, z; \pi, M, b^2, \alpha) \frac{f^*(z; \tau)}{f^*(z; \tau)}$$
$$= \log \mathbb{E}\left[\frac{f(x, y, Z; \pi, M, b^2, \alpha)}{f^*(Z; \tau)} \right]$$
$$\geq \mathbb{E}\left[\log \frac{f(x, y, Z; \pi, M, b^2, \alpha)}{f^*(Z; \tau)} \right] \quad \text{by Jensen's inequality.}$$
$$= \mathbb{E}_{f^*}\left[\log f(x, y, Z; \pi, M, b^2, \alpha) \right] - \mathbb{E}_{f^*}\left[\log f^*(Z; \tau) \right]$$

In particular, the lower bound (RHS) is closely related to the Kullback Liebler divergence $KL(\cdot)$ of $f^*(z; \tau)$ to $f(z \mid x, y; \pi, \alpha, b^2)$

$$KL\left[f^*(z; \tau) \,\middle|\, f(z \mid x, y; \alpha, \pi, b^2, M) \right] = \log f(x, y; \pi, M, b^2, \alpha) - \left\{ \mathbb{E}\left[\log f(x, y, Z; \alpha, \pi, M, b^2 \right] - \mathbb{E}\left[\log f^*(Z; \tau) \right] \right\}$$

Rearranging

$$\log f(x, y; \pi, M, b^2, M) - KL[\ |\ ] = \mathbb{E}\left[\log f(x, y, Z; \alpha, \pi, M, b^2 \right] - \mathbb{E}\left[\log f^*(Z; \tau) \right],$$

Thus the lower bound is attained when $KL[\ ] = 0$ that is when $f^*(z; \tau)$ coincides with $f(z \mid x, y; \alpha, \pi, b^2, M)$. Thus, the goal is for a family of densities described by variational parameter τ, [$n \times Q$ matrix of probabilities] we can approximate marginal likelihood by its lower bound.

Noting the lower bound as:

$$\mathcal{J}(f^*(z;\tau);\pi,\alpha,\mu,\sigma^2) = E\left[\log f(x,y,z;\alpha,\pi,\mu,\sigma^2)\right] - E\left[\log f^*(z;\tau)\right]$$

Now the point estimating equations for $\alpha, \pi, \sigma^2, \mu, \tau$ are found by maximizing $\mathcal{J}(\cdot)$ w.r.t to the model's parameters.

$$\mathcal{J}(\cdot) = E\left[\log f(x,y|z;\pi,\mu,\sigma^2)\, f(z;\alpha)\right] - E\left[\log f^*(z;\tau)\right]$$

$$= E\left[\log \underbrace{f(x|z;\pi)}_{(EQ.1)}\, \underbrace{f(y|z;\mu,\sigma^2)}_{(EQ.3)}\, \underbrace{f(z;\alpha)}_{(EQ.2)}\right] - E\left[\underbrace{\log f^*(z;\tau)}_{(EQ\,5).}\right]$$

$$= E\left[\log \prod_{i=1}^{n}\prod_{j\neq i}^{n}\prod_{g=1}^{a}\prod_{l=1}^{a}\left[f(x_{ij}|z_{ig},z_{jl};\pi_{gl})\right]^{z_{ig}z_{jl}}\right] + E\left[\log \prod_{i=1}^{n}\prod_{g=1}^{a} f(y_i|z_{ig})^{z_{ig}}\right]$$

$$+ E\left[\log \prod_{i=1}^{n}\prod_{g=1}^{a} \alpha_g^{z_{ig}}\right] - E\left[\log \prod_{i=1}^{n}\prod_{g=1}^{a} \tau_{ig}^{z_{ig}}\right].$$

After taking expt. $E(z_{ig}z_{jl}) = \tau_{ig}\tau_{jl}$ and $E(z_{ig}) = \tau_{ig}$.

$$\mathcal{J}(\cdot) = \sum_{i=1}^{n}\sum_{j\neq i}^{n}\sum_{g,l}^{a} \tau_{ig}\tau_{jl}\log f(x_{ij}|z_{ig},z_{jl};\pi_{gl}) + \sum_{i=1}^{n}\sum_{g=1}^{a}\tau_{ig}\log f(y_i|z_{ig})$$

$$+ \sum_{i=1}^{n}\sum_{g=1}^{a}\tau_{ig}\log\alpha_g - \sum_{i=1}^{n}\sum_{g=1}^{a}\tau_{ig}\log\tau_{ig}.$$

For example:

$$\frac{\partial}{\partial \alpha_g}\mathcal{J}(\cdot) = \frac{\partial}{\partial \alpha_g}\left[\sum_{i=1}^{n}\sum_{g=1}^{a}\tau_{ig}\log\alpha_g\right] \Rightarrow$$

$$\Rightarrow \sum_{i=1}^{n}\tau_{ig}\frac{1}{\alpha_g} + \frac{\partial}{\partial\alpha_g}\left[\lambda\sum_{g=1}^{a}\alpha_g - 1\right] = 0$$

$$\frac{1}{\alpha_g}\sum_{i=1}^{n}\tau_{ig} + \lambda = 0$$

$$\frac{1}{\alpha_g}\sum_{i=1}^{n}\tau_{ig} = \lambda$$

$$\alpha_g = \frac{\sum_{i=1}^{n}\tau_{ig}}{\lambda}$$

But we know that

$$\sum_{g=1}^{a}\alpha_g = \sum_{g=1}^{a}\frac{1}{\lambda}\sum_{i=1}^{n}\tau_{ig}$$

$$1 = \frac{1}{\lambda}\sum_{g=1}^{a}\sum_{i=1}^{n}\tau_{ig}$$

$$1 = \frac{1}{\lambda}\sum_{i=1}^{n}\underbrace{\sum_{g=1}^{a}\tau_{ig}}$$

$$1 = \frac{1}{\lambda}\sum_{i=1}^{n}1$$

$$1 = \frac{1}{\lambda}\cdot n$$

$$\boxed{\lambda = \frac{n}{1} = n.}$$

$$\alpha_g = \frac{1}{n}\sum_{i=1}^{n}\tau_{ig}.$$

Similary, the updating equations are:

$$\hat{\pi}_{ql} = \frac{\sum_{i \neq j} \tau_{iq} \tau_{jl} x_{ij}}{\sum_{i \neq j} \tau_{iq} \tau_{jl}}$$

$$\hat{\mu}_q = \frac{\sum_{i=1}^{n} \hat{\tau}_{iq} y_i}{\sum_{i=1}^{n} \tau_{iq}}$$

$$\hat{\sigma}_q^2 = \frac{\sum_{i=1}^{n} \sum_{q=1}^{Q} \tau_{iq} (y_i - \hat{\mu}_q)^T (y_i - \hat{\mu}_q)}{P \sum_{i=1}^{N} \sum_{q=1}^{Q} \hat{\tau}_{iq} = n}$$

(Not, EQ (9) contains typo e_q = ? Should be μ_q.)

$$\hat{\tau}_{iq} \propto \frac{\alpha_q}{\hat{\sigma}_q^2} \prod_{j \neq i}^{n} \prod_{l=1}^{Q} \left[\pi_{ql}^{x_{ij}} (1 - \pi_{ql})^{1 - x_{ij}} \right]^{\hat{\tau}_{jl}} \prod_{p=1}^{P} \exp\left\{ - \frac{(y_i^{(p)} - \hat{\mu}_q^{(p)})^2}{2 \hat{\sigma}_q^2} \right\}.$$

Different to EQ (8), as the parameters we try to estimate are never dropped. !!!

Model Selection: ICL.

So far we assumed that Q was fixed. Here, we will discuss criterion for its estimation. Here, we assume that a model with Q blocks (clusters) is random variable and that all m_Q's are equally likely. We base this criterion on the integrated classification likelihood, that is:

$$f(x, y, z \mid m_Q) = f(x \mid z, m_Q) f(y \mid z, m_Q) f(z \mid m_Q).$$

Also, we will assume that all priors are separable.

$$f(x \mid z, m_Q) = \int_{\Pi_{[Q \times Q]}} f(x \mid z, m_Q, \pi) f(\pi \mid m_Q) d\pi$$

It is assume that the conditions of Laplace integral approximation are satisfied, that is that the contribution to the integral is the greatest in the neighbourhood of MLE estimate $\hat{\pi}$ and that the influence of prior $f(\pi \mid m_Q)$ is negligable.

$$\log f(x \mid z, m_Q) \approx \log f(x \mid \hat{z}, \hat{\pi}, m_Q) - \frac{1}{2} \cdot \frac{Q(Q-1)}{2} \ln \binom{n}{2}$$

$$\log f(y \mid z, m_Q) \approx \log f(y \mid \hat{z}, \hat{\mu}, \hat{\sigma}^2, m_Q) - \frac{Q}{2} (P+1) \log(nP)$$

$$\log f(z \mid m_Q) \approx \log f(z \mid m_Q, \hat{\alpha}) - \frac{(Q-1)}{2} \log(n)$$

Combing these we obtain the ICL(mg).

$$\widehat{ICL}(m_g) = \log f(x, y, z \mid m_g, \hat{\alpha}, \hat{\pi}, \hat{M}, \hat{b}) - \frac{g}{2}(p+1)\log(np) - \frac{1}{2}\frac{Q(Q-1)}{2}\log\left(\frac{n(n-1)}{2}\right) - \frac{(Q-1)}{2}\log(n).$$

This construction is ≠ to the one given in the paper, as the criterion they propose Rewards far complexity!

Experiments. (This is summarized in Table 1)

They consider P(s) of the form $\Pi = \begin{bmatrix} \lambda & \varepsilon & \cdots & \varepsilon \\ & \ddots & & \\ & & & \varepsilon \\ & & & \lambda \end{bmatrix}$ $Q \times Q$. with $d = \frac{1}{Q}$ and $n = 150$

There 2 parameters model the distance between and between-cluster Bernoulli rates. Similarly, they consider (M_2 and M_4) and this "distance" can also capture some difficult-easy cases. They consider 43 different scenarios for these models and the d simulations per ~~model~~ case is 20. To evaluate if the clusters assignments make sense they use ARI (Adjusted Rand Index) which evaluates similarity on [0,1] scale +1 being identical and 0 being totally different. For each, simulation alg. is restarted 10 times. This

Alternative algorithms.

① Spectral Multiple View Learning (SMVL)
② Hidden Markov Random fields. (Ambroise et al. 1997).
(not sure if these 2 can model for vertex features?)

Results.

In the Figure 2 (a)(b)(c) when there is strong evidence of modular structure in the network (i.e. $d(\lambda, \varepsilon) > 0$, $d(M_2, M_{g4}) > 0$) then the ConsMix Algorithm perform better. Even SMLV and HMRF algorithms. This is of course measured in terms of ARI scores. that are called rand Errors?