# Stochastic Blockmodels meet Graph Neural Networks

Nikhil Mehta Lawrence Carin Piyush Rai

International Conference on Machine Learning (ICML) 2019, Long Beach, CA

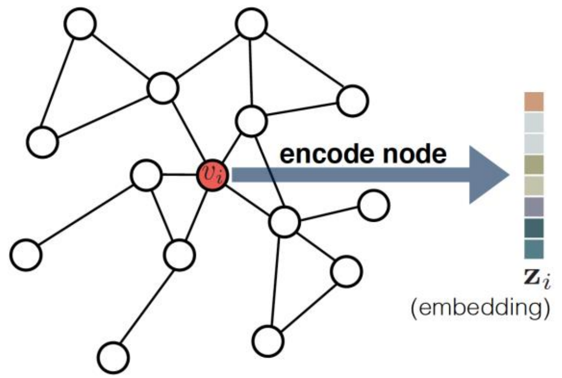
# Problem Statement

- Goal: Learn sparse node embeddings for graphs.
- Motivation:
  - Can be used for downstream machine learning tasks – link/edge prediction, node classification, community discovery.
- Some notation
  - Consider a graph associated with an adjacency matrix:

$$A \in \{0,1\}^{\{N \times N\}}$$

  - Additional side information associated with each node:

$$X \in R^{\{N \times D\}}$$



# Some Existing Work

- Probabilistic Methods:
  - A simple class of models: Stochastic Block Models (SBM) [Nowicki & Snijders, 2001]

$$z_i \sim Multinoulli(\pi) \qquad A_{\{i,j\}} \sim Bernoulli\left(z_i^T W z_j\right)$$

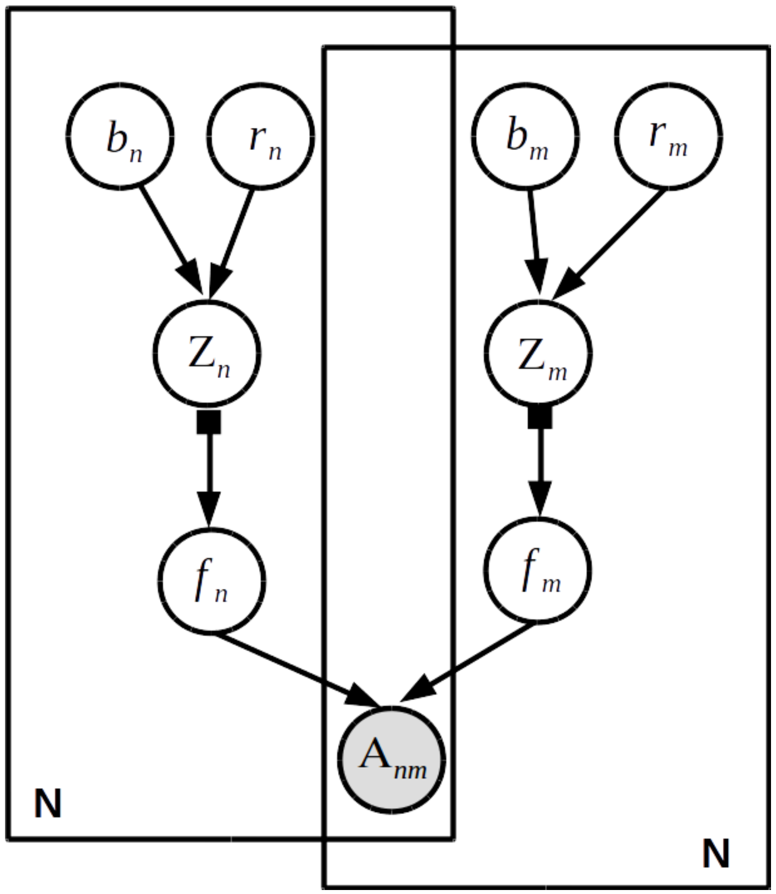
  - Overlapping SBM (OSBM) [Miller et al., 2009] – participation in multiple cochromemmunities.
    - Latent Feature Relational Model (LFRM), $z_i \in \{0,1\}^K \; K \rightarrow \infty$

$$Z \sim IBP(\alpha);\; \lambda_{\{k,k`\}} \sim \mathcal{N}\left(0, \sigma_\lambda^2\right);\; A_{\{i,j\}} \sim Bernoulli(\sigma(z_i^T \Lambda z_j))$$

  - Can handle uncertainty & missing data better. ☺
  - Interpretability can be achieved by suitable choice of prior. ☺
  - Uses iterative inference methods (MCMC, VB), not easy to scale. ☹
- What about Variational Graph Autoencoder (GVAE) [Kipf & Welling, 2016] ?
  - Encoder – Graph Convolutional Network (GCN)
  - Decoder – Link prediction: $\sigma\left(z_i^T z_j\right)$ or Node classification: softmax(g(z))
  - Fast and scalable ☺
  - Generative method + Uses deep NN = Best of both worlds? No
  - Embeddings are often not interpretable. ☹
  - What should be the size of the latent space? ☹

# Deep Generative LFRM

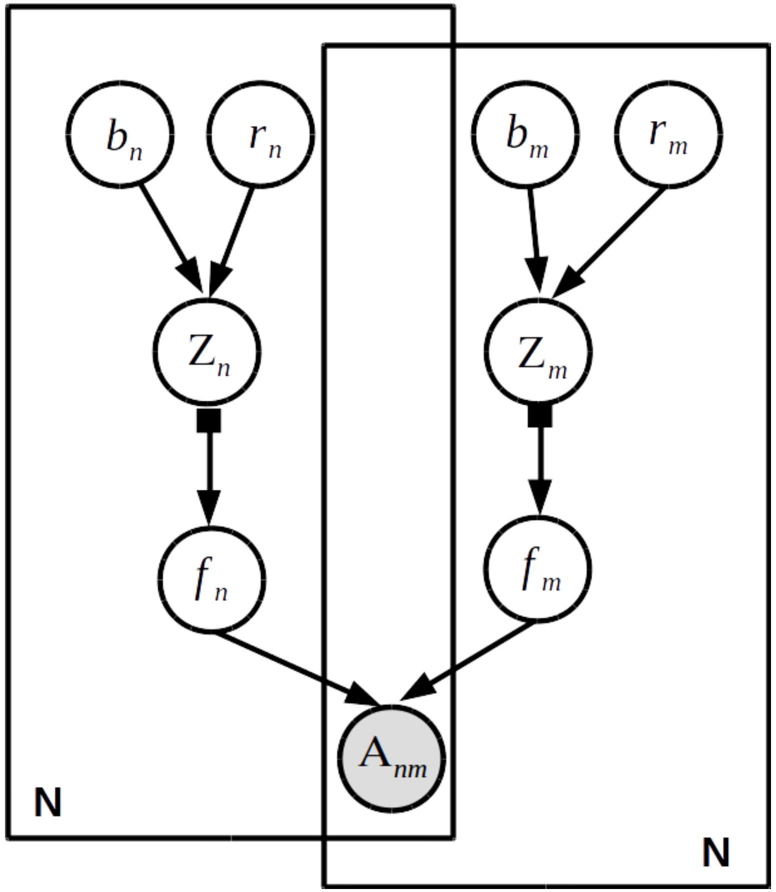
- We propose DGLFRM – Deep Generative Model for Graphs
- Unification: Interpretability of SBM + fast inference via Graph Neural Network.
- Node embedding ($z_n$) is the element wise product of two other latent variables: $z_n = b_n \odot r_n$.
- $b_n \in \{0,1\}^K$ defines the node-community memberships (cluster assignments). This allows the model to infer the "active communities" for a given ($K$).
- $r_n \in \mathbb{R}^K$ defines the node-community membership strength.



# Deep Generative LFRM

## Generative Story

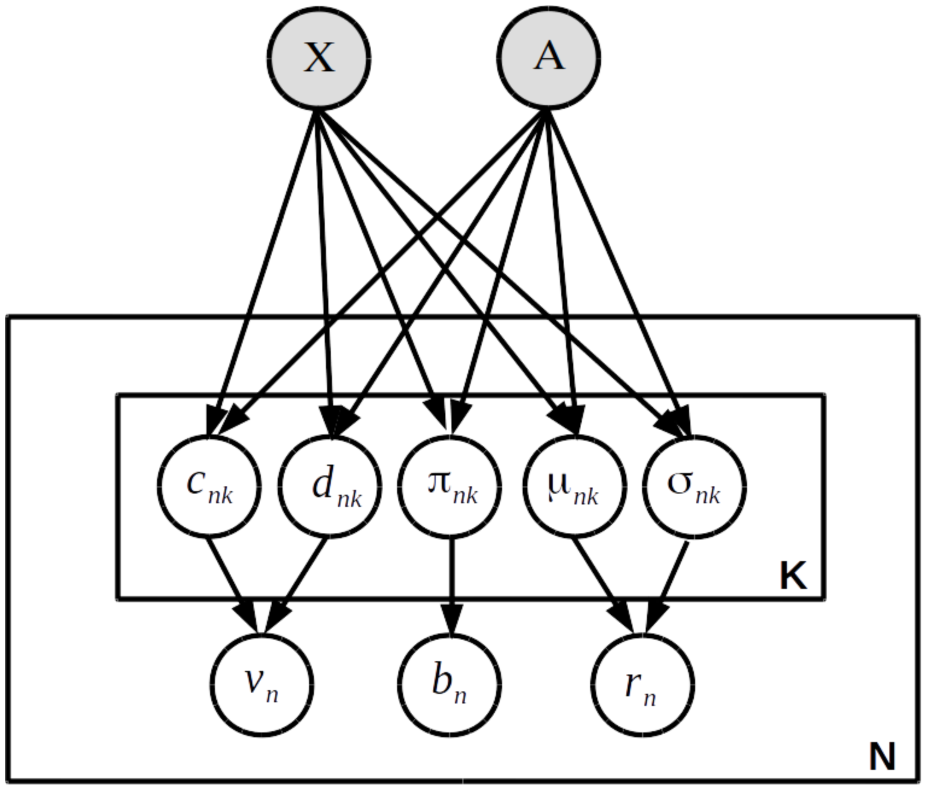
- Membership vector ($b_n \in \{0,1\}^K$)
  - Stick-breaking IBP
    - $v_k \sim Beta(\alpha, 1),\ \ k = 1,2, \ldots, K$
    - $\pi_k = \prod_{j=1}^{k} v_j,\ \ \ b_{nk} \sim Bernoulli(\pi_k)$
- Membership Strength ($r_n \in \mathbb{R}^K$)
    - $r_n \sim \mathcal{N}(0,1)$
- Node embedding: ($z_n = b_n \odot r_n$)
- $f_n = f(z_n)$, where $f$ is a multi-layered perceptron.
- $p\left(A_{\{nm\}} \middle| f_n, f_m\right) = \sigma(f_n^T f_m)$
- Posterior: $\text{p}(v, b, r | A, X)$



# Deep Generative LFRM

## Inference Network

- Full mean-field approximation: Approximate the true posterior with the variational posterior.
- $q_\phi(v, b, r) = \prod_{k=1}^{K} \prod_{n=1}^{N} q_\phi(v_{nk}) q_\phi(b_{nk}) q_\phi(r_{nk})$
  - $q_\phi(v_{nk}) = Kumaraswamy(v_{nk} | c_k, d_k)$
  - $q_\phi(b_{nk}) = Bernoulli(b_{nk} | \pi_k)$
  - $q_\phi(r_{nk}) = \mathcal{N}(\mu_n, diag(\sigma_n^2))$
- Kumaraswamy can be re-parameterized and act as a reasonable approximation for Beta. For Bernoulli, we use continuous relaxation (Concrete Distribution).



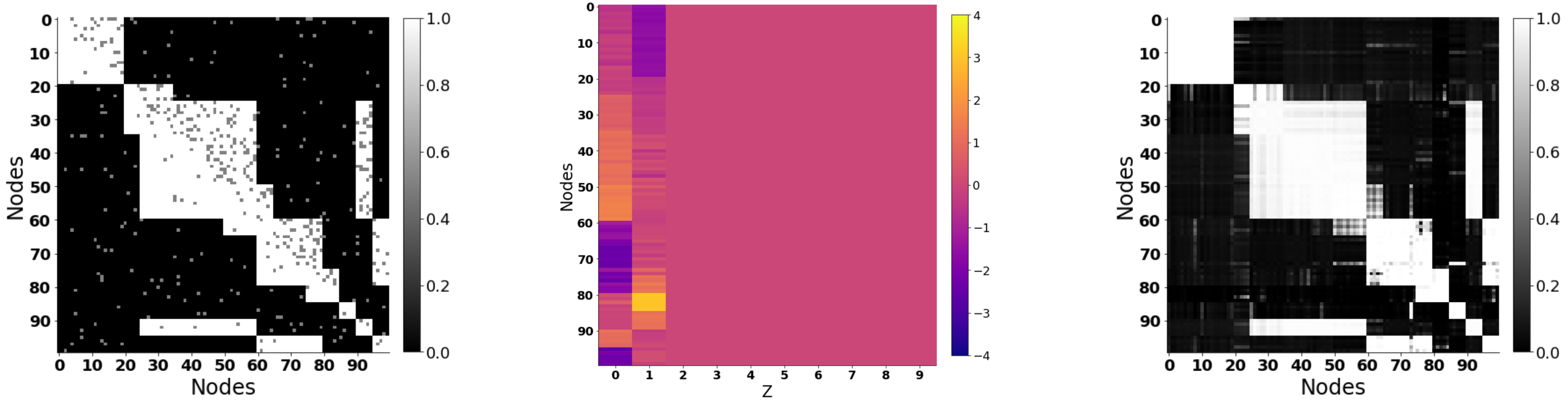
# Deep Generative LFRM

## Learning

- Since the vanilla mean-field ignores the posterior dependencies among the latent variables, we considered Structured Mean-Field: $q_\phi(v, b, r) = \prod_{k=1}^{K} q_\phi(v_k) \prod_{n=1}^{N} q_\phi(b_{nk}|v) q_\phi(r_{nk})$
- The only difference from the Mean-field approximation is that $v$ is now a global variable (same for all nodes); $\mathrm{b_{nk}|v} \sim Bernoulli(\pi_k)$.
- We can maximize the following ELBO:

$$\sum_{n=1}^{N}\sum_{m=1}^{N}(\mathbb{E}[\log p_\theta(A_{nm}|z_n, z_m)]) + \sum_{n=1}^{N}(\mathbb{E}[\log p_\theta(X_n|z_n)])$$
$$-\sum_{n=1}^{N}(KL\left[q_\phi(b_n|v_n)\middle|p_{\theta(b_n|v_n)}\right] + KL[q_\phi(r_n)|p_\theta(r_n)] + KL[q_\phi(v_n)|p(v_n)])$$

# Results




Performance on Link prediction task on five datasets.

| Method | NIPS12 | Yeast | Cora | Citeseer | Pubmed |
|---|---|---|---|---|---|
| SC | $0.9022 \pm .0002$ | $0.8440 \pm .0001$ | $0.8850 \pm .0000$ | $0.8500 \pm .0100$ | $0.8780 \pm .0100$ |
| DW | $0.8634 \pm .0000$ | $0.6699 \pm .0002$ | $0.8500 \pm .0001$ | $0.8360 \pm .0001$ | $0.8440 \pm .0000$ |
| VGAE | $0.9114 \pm .0042$ | $0.8349 \pm .0002$ | $0.9328 \pm .0001$ | $0.9200 \pm .0002$ | $0.9394 \pm .0088$ |
| LFRM | $0.8870 \pm .0000$ | $0.8268 \pm .0005$ | $0.9060 \pm .0033$ | $0.9118 \pm .0031$ | $0.9197 \pm .0054$ |
| DGLFRM-B | $\mathbf{0.9120} \pm .0021$ | $\mathbf{0.8442} \pm .0002$ | $0.9259 \pm .0023$ | $0.9153 \pm .0031$ | $0.9454 \pm .0050$ |
| DGLFRM | $0.9005 \pm .0027$ | $0.8388 \pm .0002$ | $\mathbf{0.9376} \pm .0022$ | $\mathbf{0.9438} \pm .0073$ | $\mathbf{0.9497} \pm .0035$ |

# Thank you

Please come to our poster @ 06:30PM Pacific Ballroom #180