Network Community Detection using Neural Embeddings

Chinmay Sharma

International Institute of Information Technology Hyderabad

chinmay.sharma@research.iiit.ac.in

Introduction and Motivation

- Networks have become ubiquitously used for modelling complex domains - Social Science, Transportation, Neuroscience, Biology.

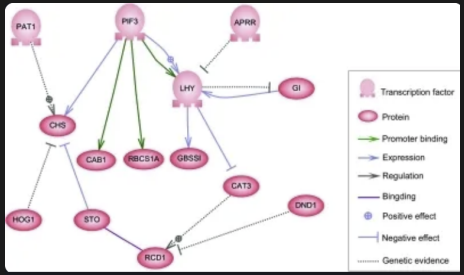

Figure: Gene Regulatory Network

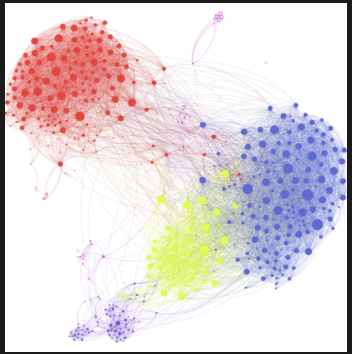

Figure: The Social Network

Introduction and Motivation

- Complex, High-Dimensional and Discrete objects $\therefore$ Non-trivial to create useful representation
- Embedding created for one domain may not generalize.

Introduction and Motivation

- Neural Networks have been used to create embeddings through methods such as node2vec and DeepWalk.
- Black-Boxed $\therefore$ lack of understanding about why these methods actually work

Introduction and Motivation

This paper's contribution:

- For the task of Community Detection, graph embedding methods based on neural networks can resolve communitites **down to Information Theoretical Limit**.

Prerequisities (1) - SBM

It is a Random Graph Model with communities.

$$SBM(n, p, Q)$$

n : Number of nodes
$p = (p_1, p_2, ..p_k)$: Probability Vector with relative size of communities
$P = \text{diag}(p)$: Diagonal matrix with p as elements

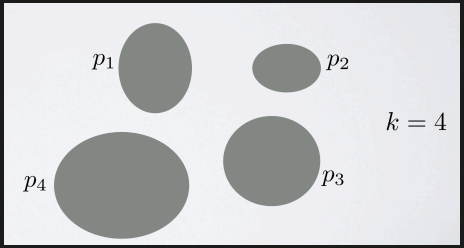

Figure: Probability vector in SBM

Prerequisites (1) - SBM

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & \cdots & Q_{1k} \\ Q_{21} & Q_{22} & \cdots & Q_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{k1} & Q_{k2} & \cdots & Q_{kk} \end{bmatrix}$$

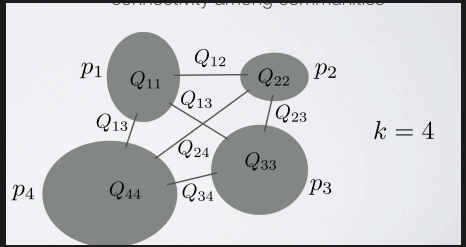

Figure: Q matrix in SBM

Prerequisites (1) - SBM Extensions

- **Profile Probability Model (PPM)**:
  - Focuses on clustering nodes based on their connection patterns while accounting for heterogeneous structures in the network.
  - Treats the probability of edge formation as influenced by a profile vector that determines the community-specific connection behavior.
- **Latent Regularized Framework (LRF)**:
  - Extends the Stochastic Block Model (SBM) by introducing latent variables to better capture complex relationships.
  - Provides a way to incorporate additional side information or regularization terms, which helps in modeling networks with more nuanced structures.

Prerequisites (2) - Graph Embeddings

- Inspired by NLP, in particular word2vec. Distributional Hypothesis: Words are known by their company
- Maximize the likelihood of the corpus. Embeddings are obtained by feeding it into a skipgram.
- Words in sentences have only 2 neighbours. How to extend it to graphs?

Prerequisites (2) - Graph Embeddings

- **Idea: Use random walkers.**

Prerequisites (2) - Graph Embeddings

- **Idea: Use random walkers.**
- Neighbourhood of word $\equiv$ Neighbours of Node.
- Sentence $\equiv$ Sequence of nodes generated by RW.
- Use a skipgram neural net to obtain embeddings.
- The difference between these methods is only in the specifics of how the random walk is done.

Prerequisites (2) - Graph Embeddings Summary

Aspect	DeepWalk	Node2Vec	LINE
Walk Strategy	Uniform random walks	Biased random walks with p and q control	Immediate and One-hop neighbourhood
Neighborhood Captured	Simple local structure	Flexible, capturing local and global structures	First- and second-order proximity
Algorithmic Complexity	Moderate	Higher due to biased walk computation	Lower (for first and second-order optimizations separately)
Scalability	Good	Good	Very good
Preservation of Proximity	Context-based similarity	Structural equivalence and homophily	Direct and neighborhood-based similarity
Hyperparameters	Number of walks, walk length, window size	p, q, number of walks, walk length, window size	None (except embedding dimensions)

Prerequisites (3) - Spectral Embeddings

Laplacian Eigenmap:

- Non-Linear Dimensionality Reduction
- Local euclidean distances and geodesic of the manifold considered.
- The smallest non-trivial eigenvectors of the graph laplacian are used to create a heat kernel for embedding.

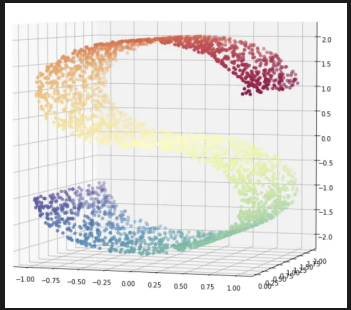

Figure: S-shaped manifold. Linear Dimensionality Reduction performs poorly

Prerequisites (3) - Spectral Embeddings

Modularity Maximization:

- A method for community detection that aims to find a partition of the network that maximizes modularity:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

  where A_{ij} is the adjacency matrix, k_i and k_j are the degrees of nodes i and j, m is the total number of edges, and $\delta(c_i, c_j)$ indicates if nodes i and j are in the same community.
- The goal is to find the partition $\{c_i\}$ that maximizes Q, indicating strong community structure.

Prerequisites (4) - Detectability Limit

For graphs generated by the PPM model, this is a limit for the mixing parameter, over which it becomes difficult to detect which communities nodes belong to. Related to information theory.
Poor performance can come from the following reasons:

- **High mixing parameter:** number of intercommunity edges is close to the number of intracommunity edges.
- **Sparse graph:** Lesser total edges $\implies$ harder to predict.

$$\mu^{*n_2 v} = \mu^* = 1 - \frac{1}{\sqrt{\langle k \rangle}}$$

Methodology

1. Generate graph embeddings using the methods described above.
2. Run K-means clustering. (Voronoi Clustering is also ran as a control Group).
3. Compare values vs Results from Spectral methods

The experiments were repeated for the PPM Model, the LFR Model and a collection of Real World Datasets over a variety of sparsity values.

Simulation Details

- **Node2Vec Parameters**:
  - Walk length: 80
  - Number of walkers per node: 40
  - Window length: 10
  - Number of training epochs: 1
  - Random walk parameters: $p = 1$, $q = 1$ (no bias)
  - Word2Vec implemented using the `gensim` package (version 4.3, default parameters)
- **LINE Parameters**:
  - Number of walks increased to 400 to compensate for fewer training iterations compared to Node2Vec.
- **DeepWalk Parameters**:
  - Number of walks increased to 120.
  - Other parameters set similar to Node2Vec.

Results

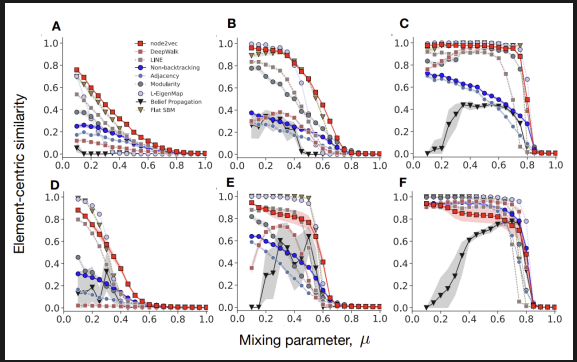

Figure: Results for LFR Model

Results

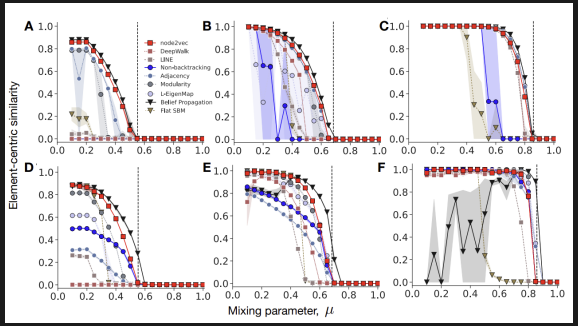

Figure: Results for PPM

Results

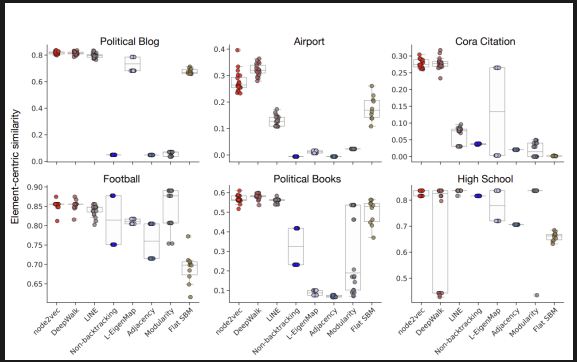

Figure: Results for Real Networks

Analytical Proof

Outline:

1. We see that all these embeddings are likelihood maximization problems. But on summing the log likelihoods we get complex non-linear terms.
2. So, under certain assumptions (which are shown to be justified for the models we are considering e.g. Poissonian Degree Distribution for PPM), we can approximate the log term to the first order.
3. We show that the matrices are similar to the graph laplacian. Because of this, the non-trivial eigenvectors are the same

Proof for Node2Vec

Embedding Objective:

$$P(x(t+\tau) = j \mid x(t) = i) = \frac{1}{Z} \exp(\mathbf{u}_i^\top \mathbf{v}_j)$$

Factorization Matrix:

$$R_{n2v,ij} = \log \frac{1}{T} \sum_{\tau=1}^{T} \frac{P(x(t+\tau) = j \mid x(t) = i)}{P(x(t) = j)}$$

Approximation for Large T:

$$R_{n2v,ij} \approx \hat{R}_{n2v,ij} = \frac{1}{T} \sum_{\tau=1}^{T} \left(\frac{P(x(t+\tau) = j \mid x(t) = i)}{P(x(t) = j)} \right) - 1$$

Proof for Node2Vec

Matrix Form:

$$\hat{R}_{n2v} = \frac{2m}{T}\left[\sum_{\tau=1}^{T}(D^{-1}A)^{\tau}\right]D^{-1} - \mathbf{1}_{n\times n}$$

Connection to Normalized Laplacian:

$$\hat{R}_{n2v} = 2mD^{-1/2}\left[\frac{1}{T}\sum_{\tau=1}^{T}(I-L)^{\tau} - \frac{D^{1/2}\mathbf{1}_n\mathbf{1}_n^{\top}D^{1/2}}{2m}\right]D^{-1/2}$$

Proof for Node2Vec

This same proof works for LINE as LINE is just node2vec with Window size T=1

Proof for Deep Walk

- **Objective Function:** DeepWalk embeds nodes by factorizing a matrix with elements:

$$R_{ij}^{\mathrm{DW}} := \log\left(\frac{1}{T}\sum_{\tau=1}^{T}\frac{P(x^{(t)}=i, x^{(t+\tau)}=j)}{P(x^{(t)}=i)\cdot\frac{1}{n}}\right)$$

- **Challenge:** The element-wise logarithm complicates the derivation of spectral properties.
- **Approximation (Large T):** As $T \to \infty$, stationary state reached:

$$\lim_{\tau\to\infty} P(x^{(t)}=i, x^{(t+\tau)}=j) \approx P(x^{(t)}=i)\cdot\frac{k_j}{n\langle k\rangle}$$

Proof for Deep Walk

- **Simplification for Poisson Degree Distribution:**

$$\frac{k_j}{n\langle k\rangle} \approx \frac{1}{n}$$

- **Taylor Expansion (Eq. (10)):**

$$R_{ij}^{\mathsf{DW}} \approx \hat{R}_{ij}^{\mathsf{DW}} = \frac{1}{T}\sum_{\tau=1}^{T} \frac{P(x^{(t)}=i, x^{(t+\tau)}=j)}{P(x^{(t)}=i)\cdot\frac{1}{n}} - 1$$

Proof for Deep Walk

Matrix Form:

$$\hat{R}^{\mathsf{DW}} = \frac{n}{T}\left[\sum_{\tau=1}^{T} D^{-1}A^{\tau}\right] - 1_{n\times n}$$

Eigenvectors: Derived from the normalized Laplacian via $D^{1/2}$ and $D^{-1/2}$.

Conclusion: DeepWalk exhibits an information-theoretical detectability limit similar to node2vec.

References

Kojaku, S., Radicchi, F., Ahn, YY. et al. Network community detection via neural embeddings. Nat Commun 15, 9446 (2024). https://doi.org/10.1038/s41467-024-52355-w