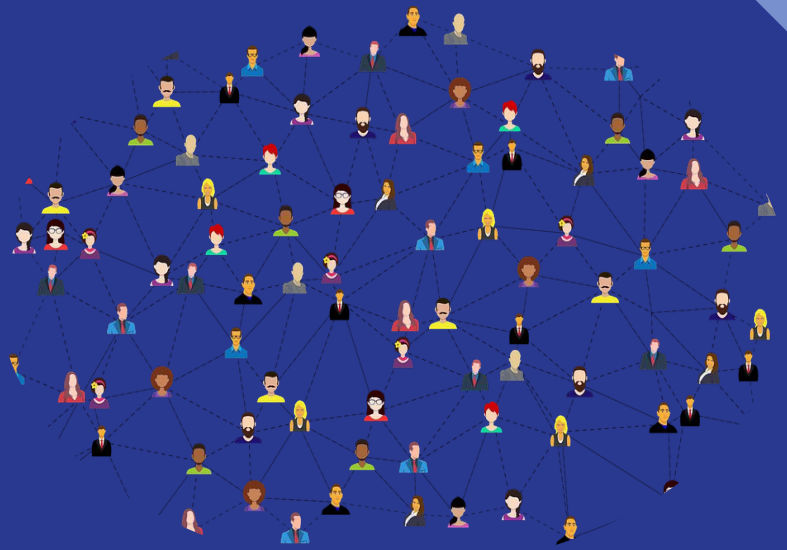
Applications of Network Embedding in Social Network Analysis

Manan Talwar, Lucy Bodtman & Justin Baltazar



Introduction

- Machine Learning (ML)
  - Application of artificial intelligence providing machines the ability to learn without explicitly being programmed.
- Graph Based Machine Learning
  - Data Clustering
  - Classification and Regression
- Social network analysis (SNA)
  - The process of investigating social structures through the use of networks and graph theory.
- Applications of SNA
  - Information spread on social networks
  - Friend and business networks
  - Disease transmission modeling
  - Data aggregation and mining

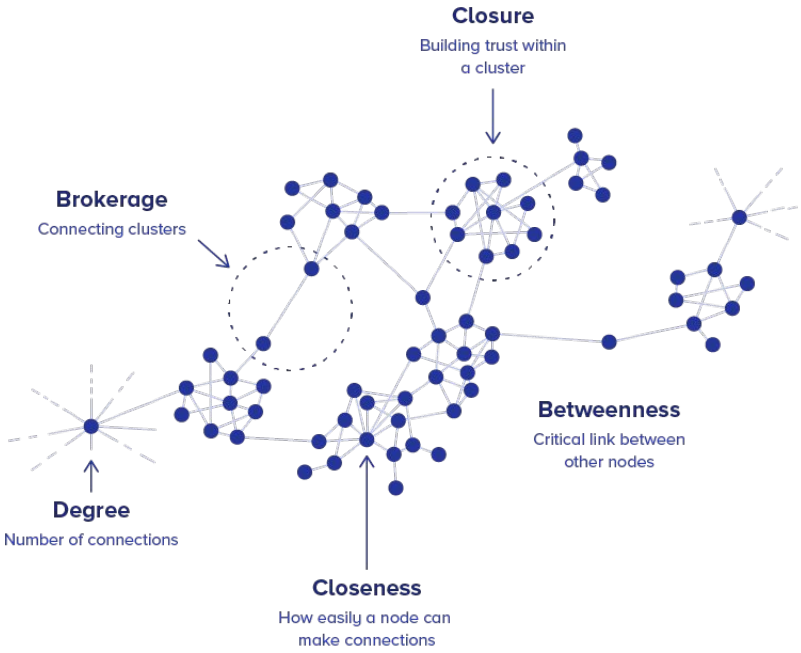


Figure 1: SNA Graph Characteristics

Introduction

- Node Embeddings
  - A transformation of nodes of a graph into a set of vectors.
  - Similarity in the embedding space (e.g., dot product) must approximate similarity in the original network.
  - Should capture the graph topology, node-to-node relationship, and other relevant information about the graph, its subgraphs, and nodes.

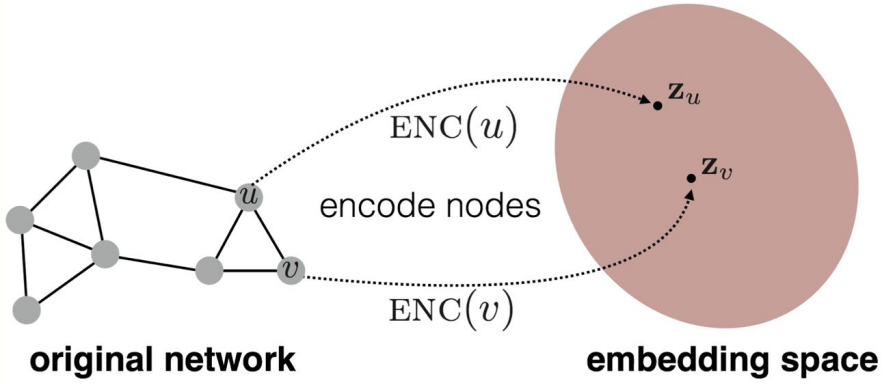


Figure 2: Network Embedding (via Stanford SNAP)

Aims

- To understand the mathematical principles behind network embedding.
- To understand the application of network embedding in graph based machine learning algorithms.
- To explore the applications of network embedding in large scale social network analysis by implementing these models on sample data.

Background: DeepWalk

- DeepWalk is a **graph neural network** that uses a **randomized path traversing** technique to provide insights into localized structures within networks.
- Starting from the target root, the algorithm randomly select a neighbor of that node, and add it to the path continue through the walk until the desired number of steps has been taken.
- This path information is used to train the **Skip-Gram algorithm** which tries to **predict in n-dimensional space the neighbors around the target node** within our network.

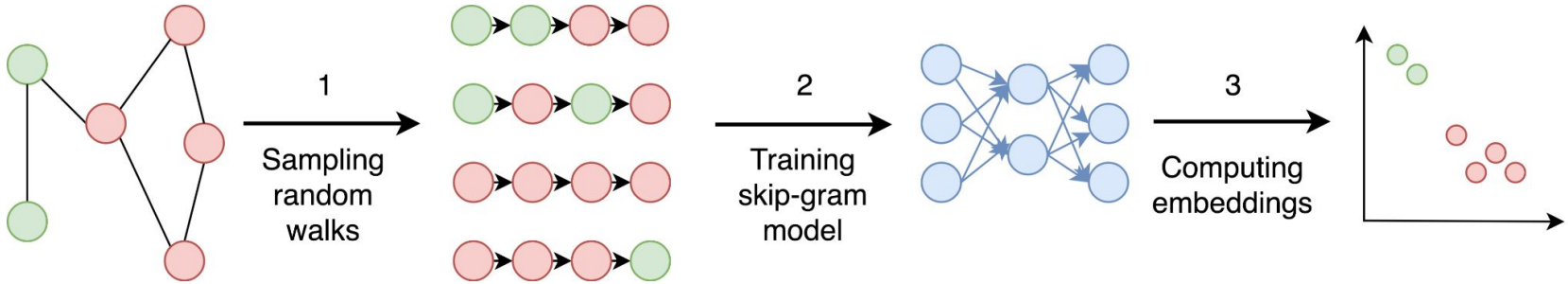


Figure 3: DeepWalk (via IIT Roorkee)

Background: Walklets

- Walklets is a procedure that uses **random walks** to approximate the **pointwise mutual information matrix** obtained by individual normalized **adjacency matrix powers**.
- Similar to DeepWalk, the network is modeled through a series of **truncated random walks** started at each node. Additionally, we choose to **skip some of the nodes** in the random walk. In this way, we form a set of relationships which are sampled from **successively higher powers of A**.
- Finally, the **skip-gram loss function** is used for optimization.

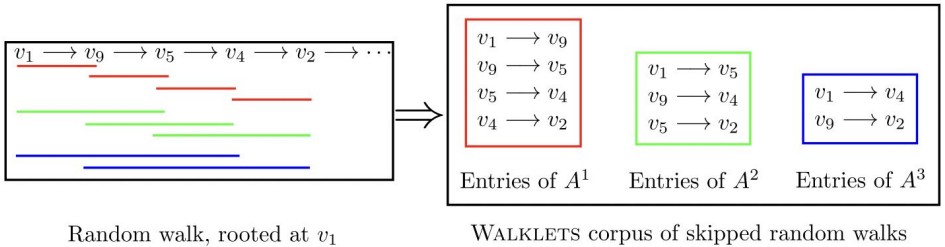


Figure 2: Overview of WALKLETS. Our method samples edges from higher powers of the adjacency matrix using a rooted random walk and skips over vertices. An edge sampled from A^k represents a path of length k in the original graph.

Figure 4: Walklets (Perozi et. al.)

Background: GLEE

- GLEE is a **linear algebraic method** that arises from **spectral graph theory**. The procedure extracts the **eigenvectors corresponding to the largest eigenvalues of the graph Laplacian**. These vectors are used as the node embedding.
- Given a graph G, consider its Laplacian matrix L. Using singular value decomposition we may write $L = SS^T$ for a unique matrix. Define S^d as the matrix of the first *d* columns of S. If *i* is a node of G , define its *d*-dimensional Geometric Laplacian Eigenmap Embedding (GLEE) as the *i*'th row of S^d.

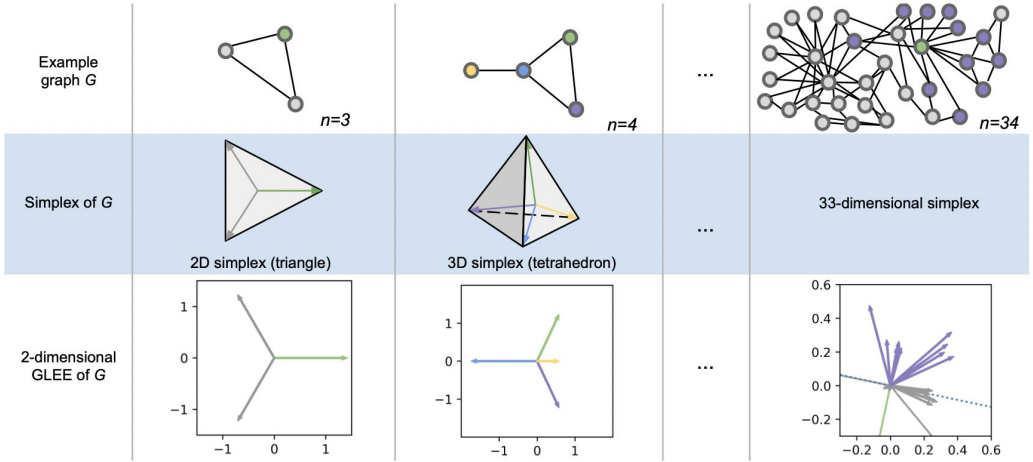


Figure 5: GLEE (Torres et. al.)

Methods

We developed implementations in Python for each of the network embedding methods described in the previous section. We used the **Karateclub API** for running the embedding models.

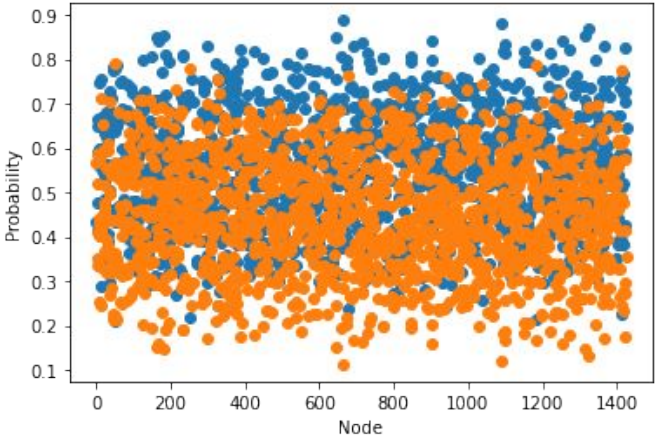
We then used **Logistic Regression** and **Random Forest classifiers** from scikit-learn to analyze three datasets available open source via **Stanford SNAP**:

- **GitHub dataset (V = 37700, E = 289003) :** A social network where nodes correspond to developers who have starred at least 10 repositories and edges to mutual follower relationships. The task is to classify nodes as web or machine learning developers.
- **Deezer dataset (V = 28281, E = 92752):** A user-user network of European members of the music streaming service Deezer. The links represent mutual friendships of the users. The task is the classification of the users' gender.
- **Twitch dataset (V = 7126, E = 35324):** User-user networks where nodes correspond to Twitch users and links to mutual friendships. The task is classification of whether a streamer uses explicit language.

Results - Twitch Dataset

Graph Embedding Method	Logistic Regression AUC	Random Forest AUC
GLEE	0.5004	0.5383
Walklets	0.5684	0.5653
Deepwalk	0.5718	0.5597

Table 1: AUC table for Twitch Dataset



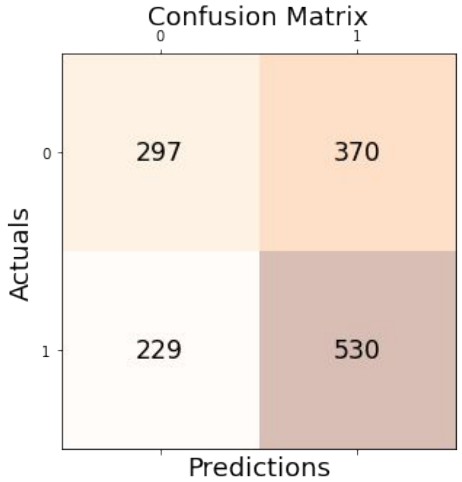
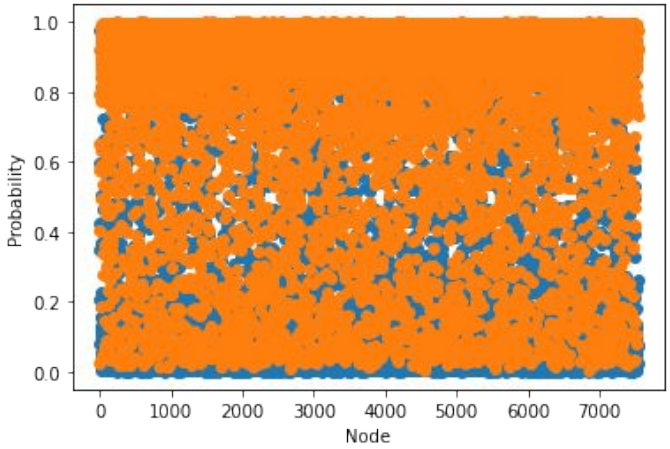


Figure 6: Predicted Probability & Confusion Matrix for best performing model

Results - Github Dataset

Graph Embedding Method	Logistic Regression AUC	Random Forest AUC
GLEE	0.5000	0.7064
Walklets	0.7821	0.7688
Deepwalk	0.7604	0.6112

Table 2: AUC table for Github Dataset



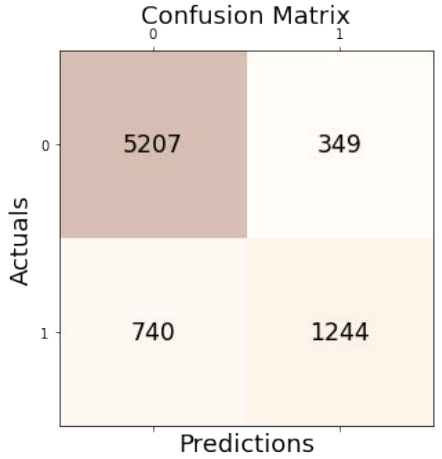
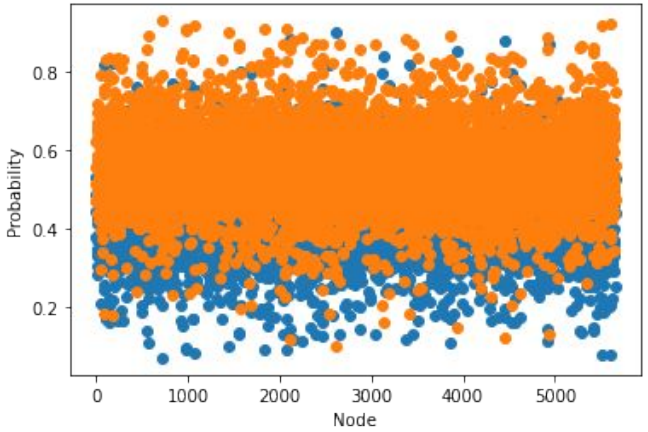


Figure 7: Predicted Probability & Confusion Matrix for best performing model

Results - Deezer Dataset

Graph Embedding Method	Logistic Regression AUC	Random Forest AUC
GLEE	0.5015	0.5357
Walklets	0.5280	0.5418
Deepwalk	0.5263	0.5095

Table 3: AUC table for Deezer Dataset



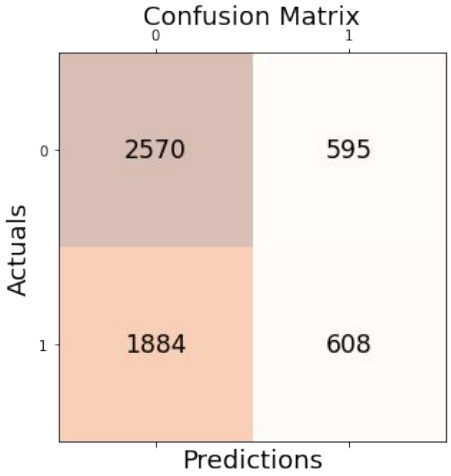


Figure 8: Predicted Probability & Confusion Matrix for best performing model

Inferences

- Random walk based methods consistently outperform linear algebraic methods in our study.
- Specifically Walklet emerges to be the best performing model in 2 of the 3 datasets.
- AUC over testing set (20% of the dataset) agree with existing literature in the discipline.
- GLEE tends to work well with ensemble Random Forests but works poorly in Logistic Regression, an observation consistent through literature we explored.

Conclusions

- We observed that none of the models are universally ideal.
  - Each network embedding method has its pros and cons.
  - There are theoretical limitations to the performance and convergence of each of these methods.
- We also note that it is crucial to understand that embeddings are approximations to the data.
  - Embeddings are lossy and hence, not all graph characteristics are preserved.
  - It is imperative that one understands the data well and select an embedding model accordingly.
  - Data preprocessing and analysis to understand its structure is the key.
- We identified the research potential in the field of SNA via network embedding
  - Our promising findings are in tandem with those reported by other researchers in this discipline.
  - We did not tune hyperparameters because it is an ongoing debate in this field. We present this as a limitation of our study.
  - We invite the academic community to explore the vast potential of node embeddings in graph based machine learning.

References

[1]

"Node Representation Learning." https://snap-stanford.github.io/cs224w-notes/machine-learning-with-networks/node-representation-learning (accessed Dec. 06, 2022).

[2]

B. Perozzi, V. Kulkarni, H. Chen, and S. Skiena, "Don't Walk, Skip! Online Learning of Multi-scale Network Embeddings." arXiv, Jun. 24, 2017. doi: 10.48550/arXiv.1605.02115.

[3]

B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online Learning of Social Representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug. 2014, pp. 701–710. doi: 10.1145/2623330.2623732.

[4]

L. Torres, K. S. Chan, and T. Eliassi-Rad, "GLEE: Geometric Laplacian Eigenmap Embedding," *Journal of Complex Networks*, vol. 8, no. 2, p. cnaa007, Apr. 2020, doi: 10.1093/comnet/cnaa007.

[5]

B. T.k., C. S. R. Annavarapu, and A. Bablani, "Machine learning algorithms for social media analysis: A survey," *Computer Science Review*, vol. 40, p. 100395, May 2021, doi: 10.1016/j.cosrev.2021.100395.

[6]

R. Ganguli, A. Mehta, and S. Sen, "A Survey on Machine Learning Methodologies in Social Network Analysis," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Jun. 2020, pp. 484–489. doi: 10.1109/ICRITO48877.2020.9197984.

[7]

M. K. Hayat *et al.*, "Towards Deep Learning Prospects: Insights for Social Media Analytics," *IEEE Access*, vol. 7, pp. 36958–36979, 2019, doi: 10.1109/ACCESS.2019.2905101.

[8]

B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-scale Attributed Node Embedding." arXiv, Mar. 21, 2021. doi: 10.48550/arXiv.1909.13021.

[9]

"Stanford Large Network Dataset Collection." https://snap.stanford.edu/data/ (accessed Dec. 06, 2022).