Benjamin Lapostolle
benjamin.jean.lapostolle@gmail.com

Lab session # 5
ALTEGRAD 2023

11/21/23

1 Task 4

Figure of the representation in 2D of the 100 most frequent nodes in the random walks1.

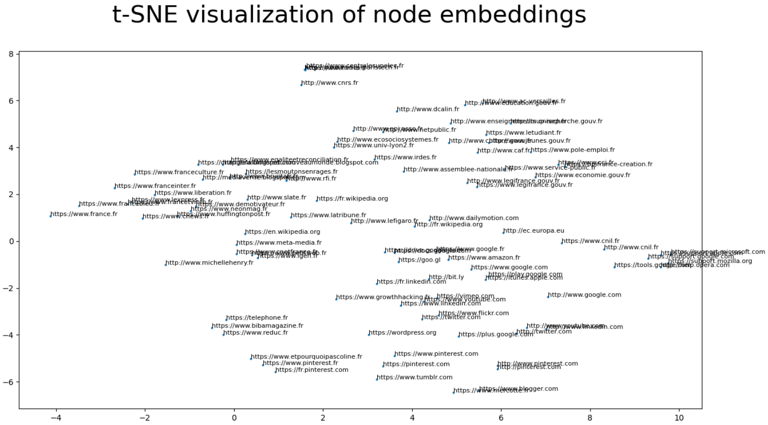

Figure 1: Representation of the 100 nodes that appear the most frequently in a 2D dimension space

2 Question 1

In the DeepWalk algorithm, embeddings of nodes are learned so that nodes frequently appearing close in the dataset of random walks have similar embeddings, maximizing the probability of a node being in the context of another node. Conversely, two nodes unlikely to appear together in random walks would have different embeddings.

Since a hierarchical softmax is employed to approximate the probability distribution, this implies that the cosine similarities of two related nodes in random walks will be high, while it will be low if they do not co-occur in the dataset.

In the case of the 2M nodes in the question, as random walks are constructed uniformly at random between the neighbors of the current nodes, starting a random walk from a K_2 component makes it very likely that the nodes inside this component will appear in each other's context. On the other hand, the probability of visiting another K_2 component is low due to the numerous components. Hence, one can conclude that the cosine similarities of the nodes within a K_2 component will be high, and the cosine similarities between nodes belonging to two distinct components will be low.

3 Task 5

Representation of the karate dataset 3.

4 Task 7

5 Question 2

Spectral Clustering Computational Complexity

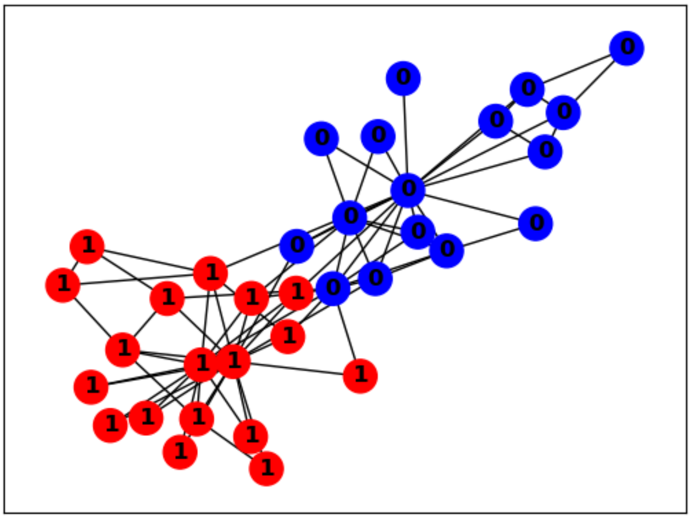

Figure 2: Representation of the karate dataset

The complexity of the spectral clustering algorithm is $O(n^3)$ because of the eigenvalue de composition. This paper explicit complexity bounds for the eigenvalue decomposition[3], and in this paper states that the general complexity is indeed $O(n^3)$ [4]. Therefore, without further assumptions or computing methods, it is difficult to apply this technique for large dataset.

Deep Walk Computational Complexity

Given the number of vertices V, the number of random walks γ, walk length t, window size w and representation size d, the time complexity of DeepWalk is dominated by the train- ing time of the Skip-gram model, which is $O(\gamma|V|tw(d + dlog|V|))$ [1].

6 Question 3

By not using directly the adjacency matrix of the graph, we cope with two issues when dealing with GNNs.

The initial step involves the normalization of matrix A. This practice is essential in neural networks to mitigate challenges such as exploding or vanishing gradients during training. Without normalization, vanishing gradients may lead to a stagnant state, while exploding gradients can result in impractical solutions as the weights grow excessively. This normalization method was first introduced in [2], in which the motivation to normalize the adjacency matrix this particular way is explained.

The second one is to add self-loops. In fact, without adding the identity to the adjacency matrix, the embedding of a node will not take into account itself. However, we would like the opposite. For instance, in a single layer network, the first row of $A \times X$, the rows of X being the embeddings of the nodes, will not be influenced by the first row of X, *ie* the embedding of the node 1. Hence, the hidden state will likely not use any information about the embedding of a node to represent it in that latent space. The quality of the classification will therefore probably be lower. In a two-layer network, the effect is more pronounced as the matrix A is used twice, and information about neighboring nodes will be utilized to classify the node.

7 Question 4

For this question, we have:

$$X = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

S_4**:**

- Adjacency matrix:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\widetilde{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\widetilde{D} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$\widetilde{D^{-\frac{1}{2}}} = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & \frac{\sqrt{2}}{2} \end{bmatrix}$$

$$\widehat{A} = \widetilde{D^{-\frac{1}{2}}}\widetilde{A}\widetilde{D^{-\frac{1}{2}}} = \begin{bmatrix} 0.25 & 0.25\sqrt{2} & 0.25\sqrt{2} & 0.25\sqrt{2} \\ 0.25\sqrt{2} & \frac{1}{2} & 0 & 0 \\ 0.25\sqrt{2} & 0 & \frac{1}{2} & 0 \\ 0.25\sqrt{2} & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

- Z_0 :

$$\widehat{A}XW_0 = \begin{bmatrix} 0.125 + 0.375\sqrt{2} & -0.15\sqrt{2} - 0.05 \\ 0.125\sqrt{2} + 0.25 & -0.1 - 0.05\sqrt{2} \\ 0.125\sqrt{2} + 0.25 & -0.1 - 0.05\sqrt{2} \\ 0.125\sqrt{2} + 0.25 & -0.1 - 0.05\sqrt{2} \end{bmatrix}$$

$$Z_0 = \begin{bmatrix} 0.125 + 0.375\sqrt{2} & 0 \\ 0.125\sqrt{2} + 0.25 & 0 \\ 0.125\sqrt{2} + 0.25 & 0 \\ 0.125\sqrt{2} + 0.25 & 0 \end{bmatrix}$$

- Z_1 :

$$\widehat{A}Z_0W_1 = \begin{bmatrix} 0.065625 + 0.084375\sqrt{2} & -0.1125\sqrt{2} - 0.0875 & 0.175 + 0.225\sqrt{2} & 0.109375 + 0.140625\sqrt{2} \\ 0.028125\sqrt{2} + 0.09375 & -0.125 - 0.0375\sqrt{2} & 0.075\sqrt{2} + 0.25 & 0.046875\sqrt{2} + 0.15625 \\ 0.028125\sqrt{2} + 0.09375 & -0.125 - 0.0375\sqrt{2} & 0.075\sqrt{2} + 0.25 & 0.046875\sqrt{2} + 0.15625 \\ 0.028125\sqrt{2} + 0.09375 & -0.125 - 0.0375\sqrt{2} & 0.075\sqrt{2} + 0.25 & 0.046875\sqrt{2} + 0.15625 \end{bmatrix}$$

$$Z_1 = \mathrm{ReLU}(\widehat{A}Z_0W_1) = \begin{bmatrix} 0.065625 + 0.084375\sqrt{2} & 0 & 0.175 + 0.225\sqrt{2} & 0.109375 + 0.140625\sqrt{2} \\ 0.028125\sqrt{2} + 0.09375 & 0 & 0.075\sqrt{2} + 0.25 & 0.046875\sqrt{2} + 0.15625 \\ 0.028125\sqrt{2} + 0.09375 & 0 & 0.075\sqrt{2} + 0.25 & 0.046875\sqrt{2} + 0.15625 \\ 0.028125\sqrt{2} + 0.09375 & 0 & 0.075\sqrt{2} + 0.25 & 0.046875\sqrt{2} + 0.15625 \end{bmatrix}$$

$$Z_1 = \begin{bmatrix} \frac{21}{320} + \frac{27}{320}\sqrt{2} & 0 & \frac{7}{40} + \frac{9}{40}\sqrt{2} & \frac{7}{64} + \frac{9}{64}\sqrt{2} \\ \frac{9}{320}\sqrt{2} + \frac{3}{32} & 0 & \frac{3}{40}\sqrt{2} + \frac{1}{4} & \frac{3}{64}\sqrt{2} + \frac{5}{32} \\ \frac{9}{320}\sqrt{2} + \frac{3}{32} & 0 & \frac{3}{40}\sqrt{2} + \frac{1}{4} & \frac{3}{64}\sqrt{2} + \frac{5}{32} \\ \frac{9}{320}\sqrt{2} + \frac{3}{32} & 0 & \frac{3}{40}\sqrt{2} + \frac{1}{4} & \frac{3}{64}\sqrt{2} + \frac{5}{32} \end{bmatrix}$$

C_4:

- Adjacency matrix:

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

$$\widetilde{A} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\widetilde{D} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

$$\widetilde{D^{-\frac{1}{2}}} = \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{3}}{3} & 0 \\ 0 & 0 & 0 & \frac{\sqrt{3}}{3} \end{bmatrix}$$

$$\widehat{A} = \widetilde{D^{-\frac{1}{2}}}\widetilde{A}\widetilde{D^{-\frac{1}{2}}} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

- Z_0 :

$$\widehat{A}XW_0 = \begin{bmatrix} 0.5 & -0.2 \\ 0.5 & -0.2 \\ 0.5 & -0.2 \\ 0.5 & -0.2 \end{bmatrix}$$

$$Z_0 = \begin{bmatrix} 0.5 & 0 \\ 0.5 & 0 \\ 0.5 & 0 \\ 0.5 & 0 \end{bmatrix}$$

- Z_1 :

$$\widehat{A}Z_0W_1 = \begin{bmatrix} 0.15 & -0.2 & 0.4 & 0.25 \\ 0.15 & -0.2 & 0.4 & 0.25 \\ 0.15 & -0.2 & 0.4 & 0.25 \\ 0.15 & -0.2 & 0.4 & 0.25 \end{bmatrix}$$

$$Z_1 = \text{ReLU}(\widehat{A}Z_0W_1) = \begin{bmatrix} \frac{3}{20} & 0 & \frac{2}{5} & \frac{1}{4} \\ \frac{3}{20} & 0 & \frac{2}{5} & \frac{1}{4} \\ \frac{3}{20} & 0 & \frac{2}{5} & \frac{1}{4} \\ \frac{3}{20} & 0 & \frac{2}{5} & \frac{1}{4} \end{bmatrix}$$

In the representation Z_1 we computed, we observe that nodes that are similar in the graph have exactly the same embedding. For the S_4 graph, only the central node has a different embedding, and in the C_4 graph, they all have the same representation since they are all similar, and that our matrix X is composed of ones for all the nodes.

8 Task 13

Figure of the task 13:

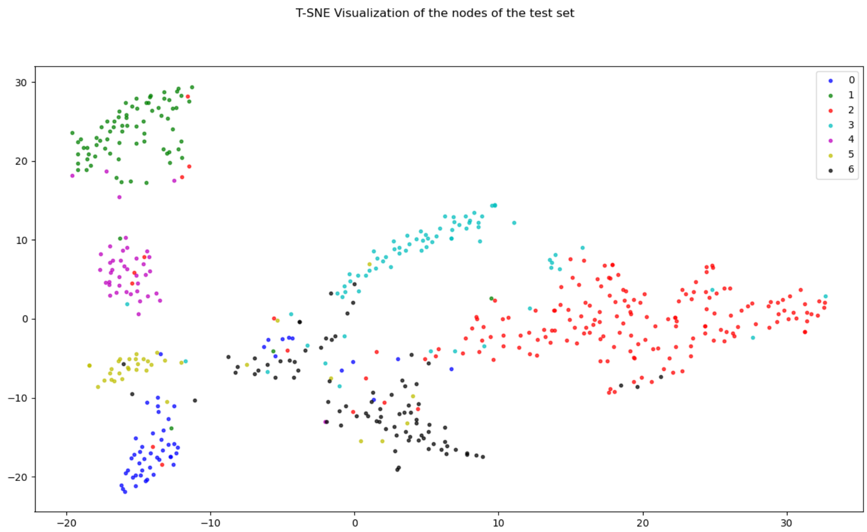

Figure 3: Representation of the karate dataset

References

[1] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. Harp: Hierarchical representation learning for networks. *Stony Brook University*, 2018.

[2] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*, 2017.

[3] Victor Y Paş and Zhao Q Chen. The complexity of the matrix eigenproblem. *Department of Mathematics & Computer Science, Lehman College, CUNY, Bronx, NY 10468*, 1994.

[4] Donghui Yan, Ling Huang, and Michael I. Jordan. Fast approximate spectral clustering. *University of California, Berkeley, CA 94720*, 2009.