Bastien LE CHENADEC
bastien.le-chenadec@eleves.enpc.fr

1 Question 1

G is a very simple graph composed of $2M$ nodes and M edges.

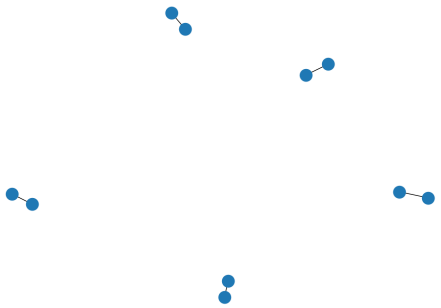

Figure 1: Graph G with $M = 5$

There is no way for the deepwalk algorithm to go from one connected component to the other. In one connected component, both edges hold symetric roles, so we expect their embeddings to be the same (cosine similarity of 1). If we take two nodes from different connected components however, they will never end up in the same random walk, so we expect their embeddings to be dissimilar (cosine similarity strictly less than 1).

2 Question 2

Complexity of Deepwalk :

In this part, N is the number of nodes in the graph, T is the length of the random walks, d is the dimension of the embeddings, K is the number of walks generated from each node, and W is the window size used in the skip-gram model.

Generating a random walk of length T starting from a node v is $O(T)$. We need to do this $N \times K$ times, so the complexity of generating all random walks is $O(NTK)$.

In the skip-gram model with hierarchical softmax, the complexity of computing the probability of a node v given a context $f(u)$ (that is, $P(v|f(u))$) is $O(\log N)$, thanks to the binary tree model used. Given one training example (one random walk), we have $O(WT \log N)$ operations. To train for a full epoch, this yields a complexity of $O(NTW \log N)$. We only took into account the forward pass, assuming that the backward pass has the same complexity.

Getting the embeddings in the case of Deepwalk is thus $O(NT(K + W \log N))$ (for a fixed number of training epochs). Assuming we chose K, T and W sensibly, independently of N, we can say that the complexity is $O(N \log N)$.

Complexity of Spectral Embeddings :

Suppose we have the adjacency matrix A of the graph G as input. In theory, computing the Laplacian matrix L is $O(N^2)$, and computing its eigendecomposition is $O(N^3)$. In practice, the method we are using for obtaining the two smallest eigenvalues implements an Arnoldi iteration [1] which allows to find the two smallest eigenvalues and the corresponding eigenvectors in $O(N^2)$ (which makes sense in terms of complexity since diagonalizing a matrix is $O(N^3)$).

Getting the embeddings in the case of Spectral Embeddings is thus $O(N^2)$.

3 Question 3

In the case of GNN, a layer of the model takes the form :

$$Z = f(\hat{A}XW)$$

What does it mean to add self-loops ? It means that A has ones on its diagonal. We can compute :

$$\begin{aligned}\hat{A}_{ii} &= \left(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\right)_{ii} \\ &= \frac{A_{ii}}{D_{ii}} \qquad \text{since } A \text{ and } D \text{ are diagonal.}\end{aligned}$$

Therefore $\hat{A}$ has non-zero values on its diagonal. Looking at the formula for Z, we see that adding self-loops amounts to allowing the model to attend to each node's own features when computing the node's next features. This must add some stability to the features across layers compared to the case where self-loops are not added, because intuitively the model can represent the identity function much more easily.

4 Question 4

Let's consider :

$$W^0 = \begin{pmatrix}0.5 & -0.2\end{pmatrix} \quad W^1 = \begin{pmatrix}0.3 & -0.4 & 0.8 & 0.5 \\ -1.1 & 0.6 & -0.1 & 0.7\end{pmatrix} \quad X = \begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

Star graph S_4 :

The adjacency matrix of S_4 is $A = \begin{pmatrix}0&1&1&1\\1&0&0&0\\1&0&0&0\\1&0&0&0\end{pmatrix}$. We can compute the normalized adjacency matrix :

$$\tilde{A} = \begin{pmatrix}1&1&1&1\\1&1&0&0\\1&0&1&0\\1&0&0&1\end{pmatrix} \quad \tilde{D} = \begin{pmatrix}4&0&0&0\\0&2&0&0\\0&0&2&0\\0&0&0&2\end{pmatrix} \quad \tilde{D}^{-1/2} = \begin{pmatrix}1/2&0&0&0\\0&1/\sqrt{2}&0&0\\0&0&1/\sqrt{2}&0\\0&0&0&1/\sqrt{2}\end{pmatrix}$$

$$\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2} = \begin{pmatrix}1/4&\sqrt{2}/4&\sqrt{2}/4&\sqrt{2}/4\\\sqrt{2}/4&1/2&0&0\\\sqrt{2}/4&0&1/2&0\\\sqrt{2}/4&0&0&1/2\end{pmatrix}$$

And then apply the GNN model :

$$Z^0 = \max\left(0, \hat{A}XW^0\right)$$

$$\hat{A}X = \begin{pmatrix}\frac{1+3\sqrt{2}}{4}\\\frac{2+\sqrt{2}}{4}\\\frac{2+\sqrt{2}}{4}\\\frac{2+\sqrt{2}}{4}\end{pmatrix} \quad \hat{A}XW^0 = \begin{pmatrix}\frac{1+3\sqrt{2}}{8} & -\frac{1+3\sqrt{2}}{20}\\\frac{2+\sqrt{2}}{8} & -\frac{2+\sqrt{2}}{20}\\\frac{2+\sqrt{2}}{8} & -\frac{2+\sqrt{2}}{20}\\\frac{2+\sqrt{2}}{8} & -\frac{2+\sqrt{2}}{20}\end{pmatrix} \quad Z^0 = \begin{pmatrix}\frac{1+3\sqrt{2}}{8} & 0\\\frac{2+\sqrt{2}}{8} & 0\\\frac{2+\sqrt{2}}{8} & 0\\\frac{2+\sqrt{2}}{8} & 0\end{pmatrix}$$

Z^0 becomes the input of the next layer :

$$\hat{A}Z^0 = \begin{pmatrix}\frac{7+9\sqrt{2}}{32} & 0\\\frac{10+3\sqrt{2}}{32} & 0\\\frac{10+3\sqrt{2}}{32} & 0\\\frac{10+3\sqrt{2}}{32} & 0\end{pmatrix} \quad \hat{A}Z^0W^1 = \begin{pmatrix}\frac{21+27\sqrt{2}}{320} & -\frac{7+9\sqrt{2}}{80} & \frac{7+9\sqrt{2}}{40} & \frac{7+9\sqrt{2}}{64}\\\frac{30+9\sqrt{2}}{320} & -\frac{10+3\sqrt{2}}{80} & \frac{10+3\sqrt{2}}{40} & \frac{10+3\sqrt{2}}{64}\\\frac{30+9\sqrt{2}}{320} & -\frac{10+3\sqrt{2}}{80} & \frac{10+3\sqrt{2}}{40} & \frac{10+3\sqrt{2}}{64}\\\frac{30+9\sqrt{2}}{320} & -\frac{10+3\sqrt{2}}{80} & \frac{10+3\sqrt{2}}{40} & \frac{10+3\sqrt{2}}{64}\end{pmatrix}$$

$$Z^1 = \begin{pmatrix} \frac{21+27\sqrt{2}}{320} & 0 & \frac{7+9\sqrt{2}}{40} & \frac{7+9\sqrt{2}}{64} \\ \frac{30+9\sqrt{2}}{320} & 0 & \frac{10+3\sqrt{2}}{40} & \frac{10+3\sqrt{2}}{64} \\ \frac{30+9\sqrt{2}}{320} & 0 & \frac{10+3\sqrt{2}}{40} & \frac{10+3\sqrt{2}}{64} \\ \frac{30+9\sqrt{2}}{320} & 0 & \frac{10+3\sqrt{2}}{40} & \frac{10+3\sqrt{2}}{64} \end{pmatrix}$$

Cycle graph C_4 :

The adjacency matrix of C_4 is $A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$. We can compute the normalized adjacency matrix :

$$\tilde{A} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad \tilde{D} = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix} \quad \tilde{D}^{-1/2} = \begin{pmatrix} 1/\sqrt{3} & 0 & 0 & 0 \\ 0 & 1/\sqrt{3} & 0 & 0 \\ 0 & 0 & 1/\sqrt{3} & 0 \\ 0 & 0 & 0 & 1/\sqrt{3} \end{pmatrix}$$

$$\hat{A} = \begin{pmatrix} 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 1/3 & 1/3 \end{pmatrix}$$

And then apply the GNN model :

$$\hat{A}X = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad \hat{A}XW^0 = \begin{pmatrix} 1/2 & -1/5 \\ 1/2 & -1/5 \\ 1/2 & -1/5 \\ 1/2 & -1/5 \end{pmatrix} \quad Z^0 = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 0 \\ 1/2 & 0 \\ 1/2 & 0 \end{pmatrix}$$

Z^0 becomes the input of the next layer :

$$\hat{A}Z^0 = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 0 \\ 1/2 & 0 \\ 1/2 & 0 \end{pmatrix} \quad \hat{A}Z^0W^1 = \begin{pmatrix} 3/20 & -1/5 & 2/5 & 1/4 \\ 3/20 & -1/5 & 2/5 & 1/4 \\ 3/20 & -1/5 & 2/5 & 1/4 \\ 3/20 & -1/5 & 2/5 & 1/4 \end{pmatrix} \quad Z^1 = \begin{pmatrix} 3/20 & 0 & 2/5 & 1/4 \\ 3/20 & 0 & 2/5 & 1/4 \\ 3/20 & 0 & 2/5 & 1/4 \\ 3/20 & 0 & 2/5 & 1/4 \end{pmatrix}$$

Comments :

- In both cases we end up with the second column of Z^1 that is null. Why is that ? If we retrace the computations, it's the negative coefficient W^1_{12} that is responsible for this. In the same fashion, the negative coefficient W^0_{12} causes the second column of Z^0 to be null.
- In more general cases, a single negative coefficient is not enough to nullify a whole column of Z^1, but it can still have a significant impact.
- We also observe similarities between the lines of Z^1. These similarities are induced by the symmetric roles of some nodes in the graph. For example, in the case of C_4, all nodes have symmetric roles, while in the case of S_4, three nodes have symmetric roles.

References

[1] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK USERS GUIDE: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*. SIAM, Philadelphia, PA, 1998.