Kyllian Asselin de Beauville
kyllian.asselin-de-beauville@polytechnique.edu

Lab session # 5
ALTEGRAD 2023

21/11/23

1 Question 1

Let G be a graph consisting of M connected components, each being a complete graph K_2. In this scenario, the DeepWalk algorithm [4] is used to embed the $2M$ nodes into a vector space.

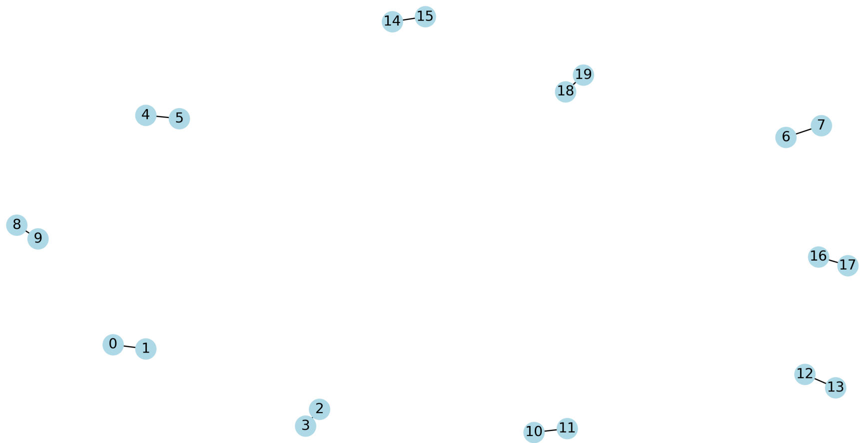

Figure 1: G with $M = 10$ connected components.

The expectation regarding the cosine similarities of the embeddings can be described as follows:

- **Within Connected Components:** For nodes within the same connected component (each K_2), the DeepWalk algorithm will generate similar random walks. This similarity arises because in a complete graph K_2, the only possible walks alternate between the two nodes. Hence, the nodes in each connected component will have very similar (if not identical) context in terms of the random walks. As a result, the cosine similarity between the embeddings of these nodes is expected to be high (close to 1), indicating a strong similarity in the vector space.
- **Between Different Connected Components:** On the other hand, for nodes in different connected components, the random walks will be entirely different, as there are no edges connecting nodes from different components. Consequently, their contexts (in terms of random walks) are entirely disjoint. The embeddings generated by DeepWalk, which rely heavily on the context in which nodes appear, will therefore be quite different for nodes in different connected components. This leads to an expectation of low cosine similarity (close to 0) between the embeddings of nodes from different connected components.

2 Question 2

According to *HARP: Hierarchical Representation Learning for Networks* [2], the time complexity of the DeepWalk algorithm is dominated by the training time of the Skip-gram model. Given the number of random walks γ, walk length t, window size w, and representation size d, the time complexity of DeepWalk is $O(\gamma|V|tw(d+d\log|V|))$, where $|V|$ is the number of vertices in the graph.

In contrast, as stated in *Approximate spectral clustering with eigenvector selection and self-tuned k* [1], the core computational component of spectral embedding is the decomposition of the graph Laplacian L, which is an $n \times n$ matrix for a graph with n nodes. This decomposition leads to a time complexity of $O(n^3)$ for spectral embedding. The cubic complexity of this method makes it computationally intensive and limits its practical applicability, especially for larger graphs.

3 Question 3

In the context of Graph Neural Networks (GNNs) [5], the absence of self-loops, i.e., working with $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ instead of $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, has a significant impact on the evolution of hidden states.

In a single-layer GNN, the absence of self-loops means that a node's features are only updated based on its neighbors' features, without considering its own features. This can result in the node features quickly becoming more similar to each other, leading to a loss of unique information inherent to each node.

In a two-layer GNN, the effect becomes more pronounced. The second layer aggregates the already modified features from the first layer, further distancing the representation from the original features of each node. This can accentuate the issues observed in single-layer GNNs, such as over-smoothing, where nodes in the same neighborhood start to have indistinguishable features, making it challenging to differentiate between them.

Overall, including self-loops in GNNs helps maintain a balance between a node's own features and the aggregated features from its neighbors, leading to more robust and discriminative node representations.

4 Question 4

Considering the GNN architecture we have been working with, which consists of two message passing layers [3], we analyze the effect of using the given weight matrices W^0 and W^1 on the star graph S_4 and the cycle graph C_4. We assume there are no biases, and the ReLU function is used as the activation function. We will compute the matrices Z^0 and Z^1 for both graphs with an all-ones matrix X as the input features.

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

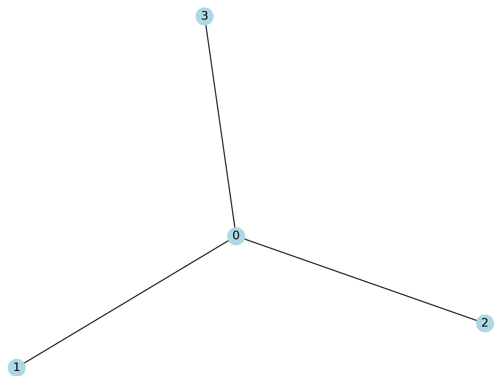

Figure 2: Star Graph S_4.

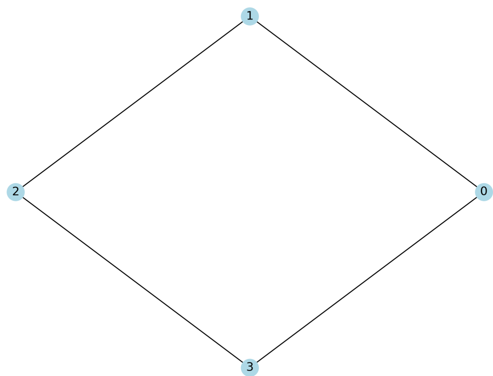

Figure 3: Cycle Graph C_4.

Star Graph S_4

The star graph S_4 (Figure 2) consists of a central node connected to three peripheral nodes. The adjacency matrix $\tilde{A}_{S_4}$ and the degree matrix $\tilde{D}_{S_4}$ for S_4 are given by:

$$\tilde{A}_{S_4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad \tilde{D}_{S_4} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \tag{2}$$

The normalized adjacency matrix $\hat{A}_{S_4}$ is computed as follows:

$$\begin{aligned}
\hat{A}_{S_4} &= \tilde{D}_{S_4}^{-\frac{1}{2}} \tilde{A} \tilde{D}_{S_4}^{-\frac{1}{2}} && (3) \\
&= \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}^{-\frac{1}{2}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}^{-\frac{1}{2}} && (4) \\
&= \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} && (5) \\
&= \begin{bmatrix} \frac{1}{4} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2\sqrt{2}} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2\sqrt{2}} & 0 & 0 & \frac{1}{2} \end{bmatrix} && (6)
\end{aligned}$$

With the normalized adjacency matrix $\hat{A}_{S_4}$, we proceed to compute the matrix $Z_{S_4}^0$, which represents the features after the first message passing layer:

$$\begin{aligned}
Z_{S_4}^0 &= \mathrm{ReLU}\left(\hat{A}_{S_4} X W^0\right) && (7) \\
&= \mathrm{ReLU}\left(\begin{bmatrix} \frac{1}{4} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2\sqrt{2}} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2\sqrt{2}} & 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0.5 & -0.2 \end{bmatrix} \right) && (8) \\
&= \mathrm{ReLU}\left(\begin{bmatrix} \frac{1}{4} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2\sqrt{2}} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2\sqrt{2}} & 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0.5 & -0.2 \\ 0.5 & -0.2 \\ 0.5 & -0.2 \\ 0.5 & -0.2 \end{bmatrix} \right) && (9) \\
&\approx \mathrm{ReLU}\left(\begin{bmatrix} 0.6553 & -0.2621 \\ 0.4268 & -0.1707 \\ 0.4268 & -0.1707 \\ 0.4268 & -0.1707 \end{bmatrix} \right) && (10) \\
&\approx \begin{bmatrix} 0.6553 & 0 \\ 0.4268 & 0 \\ 0.4268 & 0 \\ 0.4268 & 0 \end{bmatrix} && (11)
\end{aligned}$$

Having computed $Z_{S_4}^0$, we proceed to compute $Z_{S_4}^1$, which represents the features after the second message passing layer:

$$Z^1_{S_4} = \text{ReLU}\left(\hat{A}_{S_4} Z^0_{S_4} W^1\right) \tag{12}$$

$$\approx \text{ReLU}\left(\begin{bmatrix} \frac{1}{4} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2\sqrt{2}} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2\sqrt{2}} & 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0.6553 & 0 \\ 0.4268 & 0 \\ 0.4268 & 0 \\ 0.4268 & 0 \end{bmatrix} \begin{bmatrix} 0.3 & -0.4 & 0.8 & 0.5 \\ -1.1 & 0.6 & -0.1 & 0.7 \end{bmatrix}\right) \tag{13}$$

$$\approx \text{ReLU}\left(\begin{bmatrix} \frac{1}{4} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2\sqrt{2}} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2\sqrt{2}} & 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0.1966 & -0.2621 & 0.5242 & 0.3276 \\ 0.1280 & -0.1707 & 0.3414 & 0.2134 \\ 0.1280 & -0.1707 & 0.3414 & 0.2134 \\ 0.1280 & -0.1707 & 0.3414 & 0.2134 \end{bmatrix}\right) \tag{14}$$

$$\approx \text{ReLU}\left(\begin{bmatrix} 0.1849 & -0.2466 & 0.4932 & 0.3082 \\ 0.1335 & -0.1780 & 0.3560 & 0.2225 \\ 0.1335 & -0.1780 & 0.3560 & 0.2225 \\ 0.1335 & -0.1780 & 0.3560 & 0.2225 \end{bmatrix}\right) \tag{15}$$

$$\approx \begin{bmatrix} 0.1849 & 0 & 0.4932 & 0.3082 \\ 0.1335 & 0 & 0.3560 & 0.2225 \\ 0.1335 & 0 & 0.3560 & 0.2225 \\ 0.1335 & 0 & 0.3560 & 0.2225 \end{bmatrix} \tag{16}$$

We observe a distinct pattern in $Z^1_{S_4}$ where the central node (first row) has a different representation compared to the peripheral nodes. The central node's representation has higher values, indicating a higher degree of influence due to its connections to all other nodes. In contrast, the peripheral nodes (remaining rows) share identical feature representations, reflecting their similar structural roles and the fact that they are only connected to the central node. This pattern highlights the centrality of the central node and the uniformity among peripheral nodes in the star graph.

Cycle Graph C_4

The cycle graph C_4 (Figure 3) forms a closed loop with each node connected to two other nodes in a circular arrangement. The adjacency matrix $\tilde{A}_{C_4}$ and the degree matrix $\tilde{D}_{C_4}$ for C_4 are given by:

$$\tilde{A}_{C_4} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad \tilde{D}_{C_4} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \tag{17}$$

The normalized adjacency matrix $\hat{A}_{C_4}$ is computed as follows:

$$\hat{A}_{C_4} = \tilde{D}_{C_4}^{-\frac{1}{2}} \tilde{A} \tilde{D}_{C_4}^{-\frac{1}{2}} \tag{18}$$

$$= \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}^{-\frac{1}{2}} \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}^{-\frac{1}{2}} \tag{19}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{3}} \end{bmatrix} \tag{20}$$

$$= \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \tag{21}$$

With the normalized adjacency matrix $\hat{A}_{C_4}$, we proceed to compute the matrix $Z^0_{C_4}$, which represents the features after the first message passing layer:

$$
\begin{aligned}
Z^0_{C_4} &= \mathrm{ReLU}\left(\hat{A}_{C_4} X W^0\right) && (22)\\
&= \mathrm{ReLU}\left(\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0.5 & -0.2 \end{bmatrix}\right) && (23)\\
&= \mathrm{ReLU}\left(\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0.5 & -0.2 \\ 0.5 & -0.2 \\ 0.5 & -0.2 \\ 0.5 & -0.2 \end{bmatrix}\right) && (24)\\
&= \mathrm{ReLU}\left(\begin{bmatrix} 0.5 & -0.2 \\ 0.5 & -0.2 \\ 0.5 & -0.2 \\ 0.5 & -0.2 \end{bmatrix}\right) && (25)\\
&= \begin{bmatrix} 0.5 & 0 \\ 0.5 & 0 \\ 0.5 & 0 \\ 0.5 & 0 \end{bmatrix} && (26)
\end{aligned}
$$

Having computed $Z^0_{C_4}$, we proceed to compute $Z^1_{C_4}$, which represents the features after the second message passing layer:

$$
\begin{aligned}
Z^1_{C_4} &= \mathrm{ReLU}\left(\hat{A}_{C_4} Z^0_{C_4} W^1\right) && (27)\\
&= \mathrm{ReLU}\left(\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0.5 & 0 \\ 0.5 & 0 \\ 0.5 & 0 \\ 0.5 & 0 \end{bmatrix} \begin{bmatrix} 0.3 & -0.4 & 0.8 & 0.5 \\ -1.1 & 0.6 & -0.1 & 0.7 \end{bmatrix}\right) && (28)\\
&= \mathrm{ReLU}\left(\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0.15 & -0.2 & 0.4 & 0.25 \\ 0.15 & -0.2 & 0.4 & 0.25 \\ 0.15 & -0.2 & 0.4 & 0.25 \\ 0.15 & -0.2 & 0.4 & 0.25 \end{bmatrix}\right) && (29)\\
&= \mathrm{ReLU}\left(\begin{bmatrix} 0.15 & -0.2 & 0.4 & 0.25 \\ 0.15 & -0.2 & 0.4 & 0.25 \\ 0.15 & -0.2 & 0.4 & 0.25 \\ 0.15 & -0.2 & 0.4 & 0.25 \end{bmatrix}\right) && (30)\\
&= \begin{bmatrix} 0.15 & 0 & 0.4 & 0.25 \\ 0.15 & 0 & 0.4 & 0.25 \\ 0.15 & 0 & 0.4 & 0.25 \\ 0.15 & 0 & 0.4 & 0.25 \end{bmatrix} && (31)
\end{aligned}
$$

The structure in $Z^1_{C_4}$ shows uniformity across all nodes. Each node has identical feature representations, indicating a high level of symmetry in the cycle graph. This uniformity is a consequence of the regular structure of the cycle graph, where each node has the same number of neighbors (2) and is positioned similarly relative to other nodes in the graph.

References

[1] Mashaan Alshammari and Masahiro Takatsuka. Approximate spectral clustering with eigenvector selection and self-tuned k. *Pattern Recognition Letters*, 122:31–37, May 2019.

[2] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. Harp: Hierarchical representation learning for networks, 2017.

[3] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry, 2017.

[4] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '14. ACM, August 2014.

[5] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.