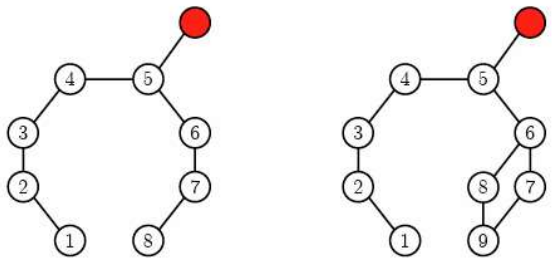
Problems

Problem 1: Effect of Depths on Expressiveness (5 points)

Consider the 2 graphs in the following figure, where all nodes have a 1-dimensional initial feature vector $x = [1]$. We use a simplified version of Graph Neural Networks (GNNs), with no nonlinearity, no learned linear transformation, and sum aggregation. Specifically, at every layer, the embedding of node v is updated as the sum over the embeddings of its neighbors ($\mathcal{N}(v)$) and its current embedding $h_v^{(t)}$ to get $h_v^{(t+1)}$. We run the GNN to compute node embeddings for the 2 red nodes respectively. Note that the 2 red nodes have different 5-hop neighborhood structure (this is not the minimum number of hops for which the neighborhood structure of the 2 nodes differs). How many layers of message passing are needed so that these 2 nodes can be distinguished (i.e., have different GNN embeddings)? Explain your answer in a few sentences.



Answer:

To distinguish the two red nodes and ensure they have different GNN embeddings, we need more than k layers of message passing, where k is the minimum number of hops in their neighborhood structure that differ.

denote the number of layers of message passing needed to distinguish the two red nodes as k. After k layers, the difference in the embeddings of the two red nodes will be at most k, which means that they will have the same embedding.

Here is the explanation:

Since the GNN model uses a simplified version with no nonlinearity, no learned linear transformation, and sum aggregation, the embedding of each node at each layer is updated by summing the embeddings of its neighbors and its current embedding.

To analyze the propagation of information through the graph, let's denote the embedding of node v at layer t as $h_v^{(t)}$, and the set of neighbors of node v as $N(v)$.

In each layer of message passing, the embedding of a node v is updated as follows:

$$h_v^{(t+1)} = h_v^{(t)} + sum\left(h_u^{(t)}\right) for\ u\ in\ N(v)$$

Start with an initial feature vector $x = [1]$ for all nodes. Therefore, the initial embedding $h_v^{(0)}$ for all nodes, including the red nodes will be [1].

To distinguish the two red nodes, their embeddings need to differ in at least one dimension. Since the initial feature vector is 1-dimensional, the embeddings of the two red nodes will differ by the sum of the values from their respective 5-hop neighborhood structures.

Problem 2: Over-Smoothing Effect (10 bonus points)

In Problem 1, we see that increasing depth could give more expressive power. On the other hand, however, a very large depth also gives rise to the undesirable effect of over smoothing. Assume we are using the aggregation function: $\boldsymbol{h}_i^{(l+1)} = \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\boldsymbol{h}_j^{(l)}$. Show that the node embedding $\boldsymbol{h}^{(l)}$ will converge as $l \to \infty$. Here we assume that the graph is connected and has no bipartite components. We also assume that the graph is undirected.

Over-smoothing thus refers to the problem of node embedding convergence. Namely, if all node embeddings converge to the same value then we can no longer distinguish them and our node embeddings become useless for downstream tasks. However, in practice, learnable weights, non-linearity, and other architecture choices can alleviate the over-smoothing effect.

Hint: You may think about the similarity between random walks and message passing. If we start at a node u and take a uniform random walk for 1 step, the expectation over the layer-l embeddings of nodes we can end up with is $\boldsymbol{h}_u^{(l+1)}$, exactly the embedding of u in the next GNN layer. Then, think about the Markov Convergence Theorem: Is the Markov chain corresponding to the message passing irreducible and aperiodic? You do not need to be super rigorous with your proof.

Answer:

To show that the node embedding $h^{(l)}$ converges as l approaches infinity, that can relate the message passing process to a random walk on the graph.

First, let's consider the message passing operation as a random walk on the graph. Starting from a node *u*, taking a uniform random walk for one step is equivalent to the aggregation step in the GNN model. The expectation over the layer-l embeddings of nodes we can end up with is $h_u^{(l+1)}$, which is exactly the embedding of u in the next GNN layer.

Now, let's analyze the properties of the Markov chain corresponding to the message passing process. We can consider each GNN layer as a step in the Markov chain. In this case, the Markov chain is irreducible because the graph is assumed to be connected, meaning that we can reach any node from any other node through a sequence of transitions. Additionally, the Markov chain is aperiodic since the graph is assumed to have no bipartite components. This means that any node can return to itself in any number of steps with a non-zero probability.

Based on the properties of the Markov chain, we can apply the Markov Convergence Theorem. This theorem states that for an irreducible and aperiodic Markov chain, the chain will converge to a stationary distribution regardless of the initial state.

In the context of the GNN, the stationary distribution corresponds to the node embeddings reaching a stable state where they no longer change significantly with further message passing. In other words, the node embeddings $h^{(l)}$ converge as l approaches infinity.

However, over-smoothing can occur when all node embeddings converge to the same value. This limits the ability to distinguish between nodes and makes the node embeddings less useful for downstream tasks. To address over-smoothing, architectural choices such as using learnable weights, non-linearities, and other methods can be employed to introduce more expressiveness and preserve the discriminative power of the node embeddings.