CS247: Advanced Data Mining (Winter 2024) February 19, 2024

Assignment 6 (Part 2)

Due Date: **March 4, 2023**

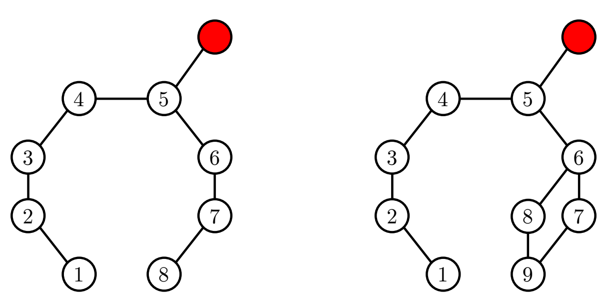
Instructions

- Submit your answer on Gradescope as a PDF file. Both typed and scanned handwritten answers are acceptable.
- Submit your solutions to Part 1 and Part 2 through GradeScope in BruinLearn separately.
- Late submissions are allowed up to 24 hours post-deadline with a penalty factor of $\mathbf{1}(t \leq 24)e^{-(\ln(2)/12)t}$.
- Ensure all sources are cited appropriately; plagiarism will be reported.

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 $\boldsymbol{h}_v^{(t)}$ to get $\boldsymbol{h}_v^{(t+1)}$. 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:

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: