Machine Learning with Graphs: Homework II - Network Science, Problems, Evaluation Due: 02/16

Spring 2023

You can choose problem 1 or 2.

Problem 1

Consider the following node classification model for attributed graphs where each node has a single attribute. For each node v, we collect the multiset of attributes within one hop from v—i.e. v and its neighbors. In the example from Figure 1, the multiset of u is $\{a, a\}$ and the multiset of v is $\{a, a, b, c\}$. Then we generate vector representations (or embeddings) in $\mathbb{R}^D$ for each vertex to represent its multiset. Finally, we learn a classifier to map the node representations to their classes. Ideally, each unique multiset should map to its own vector.

a. Given ℓ possible labels, what is the number of possible multisets for a node with degree d?

b. Assuming that each floating point in the (embedding) vector has 32 bits, how many dimensions are needed to represent these multisets from problem (a)?

c. If the graph is generated using the Barabasi-Albert model with each new node connecting to $c = 5$ existing nodes, $n \to \infty$, and $\ell = 100$, what would be a good estimate of the expected number of dimensions needed to represent at least half of the nodes? And to represent 99% of the nodes? And to represent all the nodes?

Figure 1: Problem 1.

(a) A is a true missing edge

(b) A and B are true missing edges

Figure 2: Problem 2.

Problem 2

Consider *preferential attachment* score for link prediction, where an edge is predicted based on the degrees of its endpoints. The probability $p(u, v)$ of the edge (u, v) is given as follows:

$$p(u, v) \propto deg(u)deg(v)$$

Compute the AUC for preferential attachment using the graph from Figure 2a, where A is the only true missing edge (i.e. it should be predicted by the perfect model). Do the same for the graph from Figure 2b, where both A and B are true missing edges. Do these values reflect the difference in performance between these two settings? If not, propose an alternative. You can use a library to compute the AUC. *If you are not familiar with the AUC metric, check the notes on evaluation shared on Canvas.*

Code Link: Link to code