CS 247 Homework Assignment 5, Part 1

Due date: Wednesday, Feb 28 at 11:59 PM PST
Instruction: You must write up all solutions by yourself. Start each problem on a new page, and be sure to clearly label where each problem and subproblem begins. All problems must be submitted in order (all of P1 before P2, etc.). **The Homework needs to be submitted on Gradescope** and please refer to the late submission policy in Lecture one.

1 Node Embedding (30 pts)

In this problem we explore two approaches to shallow embedding: LINE and Random Walk based embedding methods (DeepWalk). Recall that the central theme to define embedding vectors is to define a measurement of node similarity.

- In the LINE formulation, two nodes are similar if: (a) they are connected, also known as first-order proximity, or (b) their neighbors are similar, also known as second-order proximity.
- In the case of DeepWalk, two nodes are similar if they co-occur frequently in random walks. This random walk procedure is summarized as: given a starting point, randomly select a neighbor node, move to this neighbor, then repeat the process until the generated random path reaches a pre-determined length.

Suppose we want to perform node embedding for a graph $G = (V, E)$ using these two algorithms.

(a) For node embedding using LINE with first-order proximity, what's the time complexity of calculating all pairs of node similarities? How about second-order proximity? Suppose we now use a negative sampling scheme for $K << |V|$ negative samples, what's the time complexity in this case? (10 pts)

(b) Reason about why second-order proximity is required on top of first-order proximity. You may find the discussion in the original paper fruitful. (5 pts)

(c) Read the note from CS224W about Random walk-based approach and reason about comparison with LINE; in particular, the advantanges and disadvantages potentially associated with this approach. (10 pts).

(d) Reason (shortly) about the relation between Node2Vec and Word2Ve, from the perspective of sequence vs. graph data. (5 pts).

2 Knowledge Graph Embedding (15 pts)

In this problem, you are going to consider the TransE, DistMult, and RotatE models (slide09 P23-30). You are going to answer the following questions and provide reasons.

(a) If we have a knowledge graph with **friendship** and **enemy** relationship, which model(s) of the TransE, DistMult, and RotatE can we use? Please explain your reason based on the score function of each model. (**Hint**: Friendship and enemy are symmetric relationships.) (5 pts)

(b) If we have a knowledge graph with **father, grandfather, mother, and grandmother** relationship, which model(s) can we use? Please explain your reason based on the score function. (**Hint**: The father of father is grandfather. The mother of mother is grandmother. Which model(s) can model composition relationship? How?) (5 pts)

(c) For each of TransE, DistMult, RotateE, provide an example (different from part (a) and (b)) for a scenario where it cannot model the particular relationship. (5 pts)

3 Bonus: Node Embedding and its Relation to Matrix Factorization (20 points)

For this bonus problem, if you want the extra credit, you need to do some research on your own.

What to submit: For Q3.1, one or more sentences/equations describing the decoder. For Q3.2, write down the objective function. For Q3.3, describe the characteristics of W in one or more sentences. For Q3.4, write down the objective function. Recall that matrix factorization and the encoder-decoder view of node embeddings are closely related. For the embeddings, when properly formulating the encoder-decoder and the objective function, we can find equivalent matrix factorization formulation approaches.

Note that in matrix factorization we are optimizing for L2 distance; in encoder-decoder examples such as DeepWalk and node2vec, we often use log-likelihood as in lecture slides. The goal to approximate A with $Z^T Z$ is the same, but for this question, stick with the L2 objective function.

3.1 Simple matrix factorization (5 points)

In the simple matrix factorization, the objective is to approximate adjacency matrix A by the product of embedding matrix with its transpose. The optimization objective is $\min_Z ||A - Z^T Z||_2$.

In the encoder-decoder perspective of node embeddings, what is the decoder? (Please provide a mathematical expression for the decoder)

3.2 Alternate matrix factorization (5 points)

In linear algebra, we define bilinear form as $z_i^T W z_j$, where W is a matrix. Suppose that we define the decoder as the bilinear form, what would be the objective function for the corresponding matrix factorization? (Assume that the W matrix is fixed)

3.3 BONUS: Relation to eigen-decomposition (5 points)

Recall eigen-decomposition of a matrix (link). What would be the condition of W, such that the matrix factorization in the previous question (2.2) is equivalent to learning the eigen-decomposition of matrix A?

3.4 Multi-hop node similarity (5 points)

Define node similarity with the multi-hop definition: 2 nodes are similar if they are connected by at least one path of length at most k, where k is a parameter (e.g. $k = 2$). Suppose that we use the same encoder (embedding lookup) and decoder (inner product) as before. What would be the corresponding matrix factorization problem we want to solve?