CS 768: Learning With Graphs **Autumn 2020-2021**

Lecture 3: Examples of Aggregate and Per-Node Sampling

Instructor: Prof. Abir De *Scribe: Shubham Sharma and Jennish Dharma J J*

In the last lecture, we had an overview of Aggregated Sampling and Per-Node sampling. In this lecture, we'll see some applications and examples of both these sampling. In section 3.1, we'll again visit the necessity for train test split. In section 3.2, we'll see the methods and need to split into train and test set. In section 3.3, we'll see the application of both Aggregate and Per-Node Sampling. Section 3.4 contains some examples of both Aggregate and Per-Node Sampling.

3.1 Necessity for Train Test split

In Link Prediction, our task is to rank the non-edges as the graph evolves. In most cases, we only get a snapshot a graph and we can't wait for the graph to evolve to see which turns out to be edges to evaluate the model we had trained. Therefore, we have to evaluate the Link Predictor using the current snapshot of the graph itself and to do this, we split the graph into test and train edges and non edges.

3.2 Train and Test Split

To obtain a train and a test set from a given graph, we choose a fraction of edges from the graph to make a train set and remaining edges to form a test set. Similarly, we do the same for non edges. Non edges are also split into train and test so that we don't use any information about the non edges that we are going to evaluate against later. Therefore, while training a link predictor using the training set, we have to make sure that we use only the labels of the non edges other than the ones in test set and not use any information of the no edges in test set. It is not possible to not use information form the test set. For example, when we construct features for each node f_u to train the LP, we have to know the relationship between 2 nodes and thus we have to incorporate the fact that it is not an edge and thus information leaks from test set. But since the fraction of non edges in test set is much smaller than that is training set we can neglect the effects caused by this in features f_u.

3.3 Application

1. Applications of Aggregrate Sampling
    - Predicting which two authors are going to collaborate in a particular topic from citation graphs.
    - Predicting which k connections are potentially lethal in a social network.
    - Understanding reaction mechanisms.
    - Predicting interactions in Protein Protein Interaction (PPO) network
2. Applications of node specific sampling
    - Used when we want to do well across all nodes
    - Recommending k users to an user to send friend requests in Facebook
    - Recommending m movies to an user in Netflix

3.4 Examples

Let G be the graph with V={W,V,X,Y,U,Z} and E={WV,VX,XY,YU,UZ,ZW,UV} are the given edges as shows in the figure 3.1. All other links are the non-edges that are denoted by the dotted lines. In the link prediction task, ideally, we want to rank the non-edges. So, there are different likelihoods across these dotted lines and we want to rank them. Let us now try to split the graph into train and test set.(Note that we are splitting as 50%Train + 50%Test, so we can consider the splits in the below subsections as either train or test)

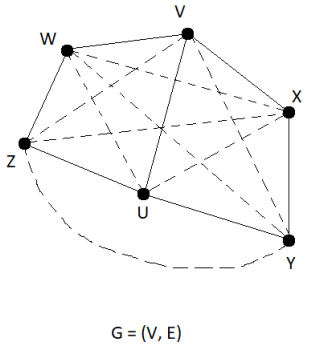

Figure 3.1: Graph

3.4.1 Aggregate Sampling(50%Train + 50%Test)

1. We spilt both edges and non-edges into 50%Train + 50%Test.
2. Here the Graph 3.2 is the train set with lines as edges and dotted lines as non edges. All other connections are what we want to predict with our LP Algorithm.
3. However, in case of features extraction of a particular node, there would be problem with the nodes like X, so, one way is we can assume that test non-edges are the training non edges, but test edges are training non edges.

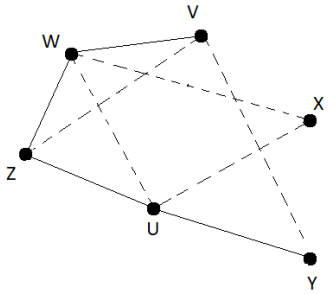

Figure 3.2: Aggregate Sampling

3.4.2 Per-Node Sampling(50%Train + 50%Test)

1. We spilt both edges and non-edges into 50%Train + 50%Test for each node in this case.
2. Graph 3.2 shows how we can split edges and non-edges for each node. It contains the split for nodes X,U,V. We can find the split for other nodes similarly.
3. In cases like let say Y has selected U as a node with edge while training, but U has not selected Y as a node with edge while training, then we can choose randomly whether or not that edge UY should be in training set or not.

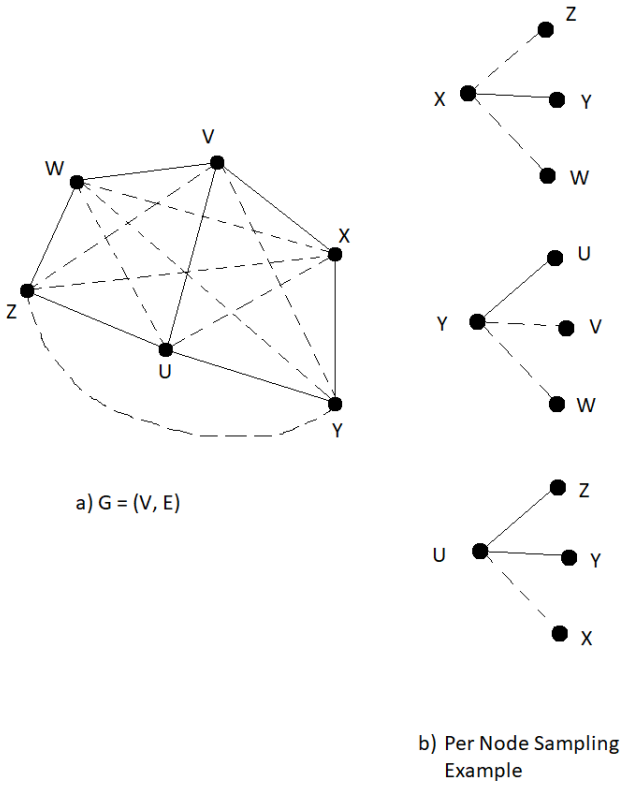

Figure 3.3: Per Node Sampling