EE226 Massive Data Mining

By: Li Yufan (518030910381)
Tang Ziang (518021910020)
Shen Zhen (518021910149)

Proposal

April 26, 2021

1 Introduction

Node classification, where each node in the dataset is assigned one or more class labels, is one of the most popular tasks for graph neural networks. With the given labels, one can extend the labeling to other nodes so that all nodes can have more information attached to them. This project is a node classification problem and we need to use the data provided to form an academic network(graph) and then train a classifier to predict in which conference the author published papers. Authors and papers in this project can be seen as 'node' while conferences can be regarded as 'label'.

The classification problem can be solve by machine learning. However, many machine learning algorithms need real-valued vectors as input while a graph is a discrete structure which consists of many nodes and edges representing some entities or relations. Besides, nodes and edges in a graph may have some attributes to describe the relations in more detail. One technique to solve this is network embedding [1], which can be used to transform nodes and edges as well as their features to vector space. And the basic principle is to learn representation for the nodes such that the relation of the nodes in the network can be preserved as much as possible.

Since there may be multiple classes of nodes or edges in a network, heterogeneous network embedding can be used for more straightforward representation of the network. In this project, we will use matapath2vec based method to apply heterogeneous embedding architectures to the network and get an efficient representation of the network for further classification task using a logistic regression classifier.

2 Related Work

2.1 Node2Vec

Node2vec [2] is an algorithm framework for network embedding and generalizes prior work DeepWalk. It uses a 2nd order biased random walk approach to generate (sample) network neighborhoods for nodes, and optimizes a graph-based objective function using SGD which maximizes the likelihood of preserving network neighborhoods of nodes.

The objective function is obtained by extending the Skip-gram architecture from language modeling to networks. It models the conditional likelihood of every source-neighborhood node pair as a softmax unit parameterized by a dot product of their features, and negative sampling is used to reduce the computational cost.

Node2vec employs a flexible biased random walk procedure that can explore neighborhoods in a BFS as well as DFS fashion. In-out and return parameters are introduced to control how fast the walk explores and leaves the neighborhood of starting node u, thus reflecting both structural equivalence and homophily between nodes.

Node2vec is also scalable. As is the same with DeepWalk, it is an online learning algorithm which builds useful incremental results, and is trivially parallelizable.

2.2 Skip-Gram

Skip-Gram [3] is initially implemented in Word2Vec in NLP and then extend to other fields both of homogeneous and heterogeneous network. Specifically, as for the problem, the objective of the Skip-Gram algorithm for a heterogeneous network is to maximize the probability of having the context $N_{\mathrm{t}}(v)$, given one node v. The optimization function is

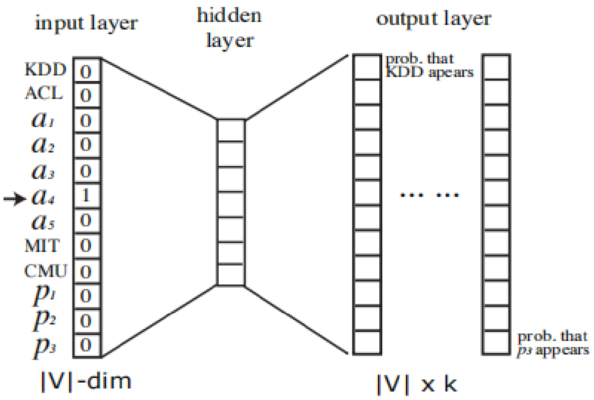
$$\arg\max_{\theta} \sum_{v \in V} \sum_{t \in T_V} \sum_{\in \in N_t(v)} \log p\left(c_t \mid v; \theta\right) \tag{1}$$

where $N_{\mathrm{t}}(v)$ denotes the neighbourhood of node v, and t gives us the type of the node. The probability $\log p\left(c_t \mid v; \theta\right)$ often presents in the form of a softmax function:

$$p\left(c_t \mid v; \theta\right) = \frac{e^{X_{c_t} \cdot X_v}}{\sum_{u \in V} e^{X_u \cdot X_v}} \tag{2}$$

where X is the representation matrix with vector for nodes as rows.By optimization, parameters gets iteration. To increase the effectiveness, negative sampling is applied to get the most of the iteration efficiency.

The whole process is shown in Fig. 1.



(b) Skip-gram in *metapath2vec*, node2vec, & DeepWalk

Figure 1: The Process of Skip-Gram [3]

2.3 Metapath2vec

Metapath2vec [3] is an algorithm focusing on representation learning for heterogeneous networks, involving diverse types of nodes. The goal of metapath2vec is to maximize the likelihood of preserving both the structures and semantics of a give heterogeneous network.

Metapath2vec includes three steps. First, heterogeneous DeepWalk is applied to heterogeneous networks to get the information of neighbourhoods with different types of nodes preserving the semantics most. Then, heterogeneous Skip-Gram is implemented to generate the model for a representation output. Finally, negative sampling is applied to increase the speed and efficiency of parameter iteration, which is so-called metapath2vec++.

Metapath2vec successfully formalizes the problem of network embedding in heterogeneous networks, which shows frequently in the scene of academic networks and social network. It can automatically discovery the structures and semantics of heterogeneous networks and present them in a matrix. The representation features then be used in further study of data mining, like node classification, clustering and link prediction.

2.4 Logistic Regression

There are two common methods to perform multi-class classification using the binary classification logistic regression algorithm: one-vs-all and one-vs-one. Assume a classification for C classes. In one-vs-all, we train C separate binary classifier for each class and run all those classifiers on any new example x we want to predict and take the class with the maximum score. In one-vs-one, we train C choose 2 classifiers = C(C-1)/2 one for each possible pair of class and choose the class with maximum votes while predicting for a new example.

Another choice is the multinomial logistic regression. Multinomial logistic regression is a form of logistic regression used to predict a target variable have more than 2 classes. It is a modification of logistic regression using the softmax function instead of the sigmoid function. The softmax function squashes all values to the range [0,1] and the sum of the elements is 1.

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{3}$$

Cross entropy is a measure of how different 2 probability distributions are to each other. If p and q are discrete we have:

$$H(p, q) = -\sum_{x} p(x) \log q(x) \tag{4}$$

This function has a range of [0, inf] and is equal to 0 when p=q and infinity when p is very small compared to q or vice versa. For an example x, the class scores are given by vector z=Wx+b, where W is a C×M matrix and b is a length C vector of biases. We define the label y as a one-hot vector equal to 1 for the correct class c and 0 everywhere else. The loss for a training example x with predicted class distribution y and correct class c will be:

$$\text{loss} = H(y, \hat{y}) = -\sum_{i} y_i \log \hat{y_i} = -\log \hat{y_c} \tag{5}$$

As in the binary case, the loss value is exactly the negative log probability of a single example x having true class label c. Thus, minimizing the sum of the loss over our training examples is equivalent to maximizing the log likelihood. We can learn the model parameters W and b by performing gradient descent on the loss function with respect to these parameters.

3 Research Plan

This project will be conducted between April 28 and June 9, 2021, for a total of six weeks. The timescale is as follows:

- 4.28 ∼ 5.5 Analyse the data and form a graph.
- 5.5 ∼ 5.19 Apply metapath2vec based method to transform the graph so that it can be the input of the classifier.
- 5.19 ∼ 6.2 Use logistic regression classifier to predict the nodes' labels and optimize the previous method to improve the classification result.
- 6.2 ∼ 6.9 Reorganize our work prepare for the short presentation. And during the whole project, we will work on the project report.

4 Expected outcome

For the first stage, we expect to get a graph that contains the full information of the data and then in the second and third stages where we will get a representation of the graph and classify the nodes, we hope to get a classification accuracy of at least 70%.

5 Options

In the case we could not get the expected results as mentioned above at the time point, we would turn to alternatives of heterogeneous networks. Homogeneous network is our first thought. By exploring potential connections between authors and papers, we could generate a homogeneous network. After that, similar algorithms for homogeneous network embedding such as Nod2vec would be a selection of us, simply because we choose metapath2vec for the heterogeneous network based on DeepWalk and Skip-Gram. Finally, we would still implement Logistic Regression to fulfill the classification task.

References

[1] Nino Arsov and Georgina Mirceva. Network Embedding: An Overview. *arXiv e-prints*, page arXiv:1911.11726, November 2019.

[2] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *KDD*, page 855–864, 2016.

[3] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, page 135–144, 2017.