Community Detection Using Deep Learning - Literature Review

Abhishek Kumar Singh
Computer Engineering Department
Delhi Technological University
Delhi, India
abhishekkumarsingh_2k18co021@$dtu.ac.in$

Prince Yadav
Bio Technology Department
Delhi Technological Univeristy
Delhi, India
princeyadav_2k19bt503@$dtu.ac.in$

***Abstract*—Community detection is a method which differentiates feature and connections of group members from that of other community members, and this is called network analysis. Apart from statistical inference processes and classic spectral clustering,deep learning techniques have also been developed for sake of community detection. We will be using K-Means and auto-encoder to improve the accuracy and reduced the problem of high computation complexity, poor parallelizability.**

I. Introduction

Everyone nowadays is connected through social media sites such as Facebook, Twitter, and Reddit. Every day, these large corporations generate enormous amounts of data. This data must be analysed in order to obtain information and establish policies for their users. Communities are present from simple datasets to real-world scenarios. So, for the facilitation of collection of data on a cluster and group community detection is necessary.

II. Related work

A major part of the computer science, mathematics and statistics community has been working on detecting a community structure within social networks[8,9,10]. Any graphical structure having its vertices interacting with one or more than one vertices can be seen as a community of users. These are following traditional approaches were used:

A. Graph Partitioning

Divides the graph into groups of g of pre-defined sizes, such that total edges are greater than total links between groups of the community.

B. Hierarchical Clustering

Graphs can contain a sequential structure, i.e. each community can be grouped into smaller groups at different levels. In such cases, successive collection strategies can be used to identify the multi-segment social structures. A score generated from the similarity between nodes is used in hierarchical integration methods. For this method to work, This algorithm must know the pre-decided community counts and their sizes.

C. Partitional Clustering

Partitional clustering divides the database into a fixed count of k groups not exceeding. The aim is that we are able to divide the data points into groups of k in order to reduce / maximize cost work based on the process of variance between areas. Other widely used cost functions are k median minimum, k-clustering sum, k-clustering, and k-centre.

D. Spectral Clustering

This method comprises all techniques that use an eigen vector matrix.Then this matrix is used to divide a collection of data points according to two similarities between them.

E. Evolutionary Algorithm

Methods are classified into two types based on single and multi-objective optimisation. Evolutionary algorithms are a type of metaheuristic optimization algorithm. They are capable of effective local learning and global searching. They can be categorized into two sets i.e. single and multi-objective optimization.

III. Failure of traditional approaches

There can be many reasons for failure of methods but we have highlighted some important once:

A. High Computation Complexity

A social network may contain a large number of nodes and many links between them. So, the dataset will have a very high degree. As we have to process this graph many times, this high degree will lead to a high computing complexity. We will require very high computation power to process this graph dataset.

B. Poor Parallelizability

For large datasets, distributed and parallel computing are very important methods for processing the data. They are depicted in the adjacency matrix and it is difficult to design algorithms for distributed and parallel computing. As data can be very large, parallelization will play an important part in computation. But as graphs are depicted in adjacency matrices it is very difficult to design distributed and parallelization algorithms.

C. Limited use of ML and deep learning

All graphs are depicted in the adjacency matrix but they are not applicable to directly feed into ML models. We have to reduce the dimensionality before feeding it into deep learning or ML or deep learning models.

IV. Major Challenges

- **Dimensionality**: A graph is a very complex structure and we can not directly feed it into our deep learning or ML model. We have to use a dimensionality reduction method before processing it in our model.
- **Scalability**: Real-world social network datasets contain a lot of nodes and edges, So we have to select a technique that will not only give good results but also scale to large datasets and perform equally good on that dataset.
- **Network Dynamics**: A real-world social network is not static. It's changing every second due to regular interaction between nodes. These changing dynamics can change network topology. The addition or deletion of a node or an edge may not only change the local community structure but also change the complete graph community structures to tackle this problem we need to learn the latest information about the dynamics of the network.

V. Proposed Approach

Auto-Encoders[2] will be used. The Auto-Encoders neural architecture is made up of two neural networks. The encoder is a neural network that converts input into low-level dimensional depictions. It also include Decoder which is a neural network that attempts to recreate data from this low-level dimensional depiction to original data. It is trained by minimising the error, depending upon the input and regenerated output. It's an unsupervised method. Auto-Encoders consist of 4 main parts:

- **Encoder**: The model is trained to learn to compress input data by reducing input dimensions into an encoded depiction.
- **Bottleneck**: In this, the encoded depiction is stored in this layer. The input data has been reduced to its smallest possible dimensions.
- **Decoder**: The model is trained to recreate data from the compressed depiction as near to the original data.
- **Reconstruction Loss**: In this process decoder's performance is measured by seeing the proximity of output to the original input.

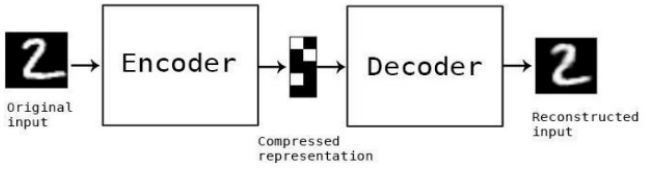

Fig. 1. Auto Encoder Architecture

To reduce the reconstruction loss the model is trained using back propagation.

We will use the K-Means which is an unsupervised technique trying to learn patterns from the input data. The target is dividing similar data into groups such that whole data can be divided into different clusters. K means trying to divide the data into K non-overlapping parts or clusters. Each data point of dataset will belongs to a subgroup. The process' of working can be understood as:

- K data points are randomly initialized.
- Each data point is categorized to its closest mean.
- The mean coordinate is updated.
- The same process is repeated for a fixed number of epochs.
- Finally, K clusters are formed after the completion of epochs.

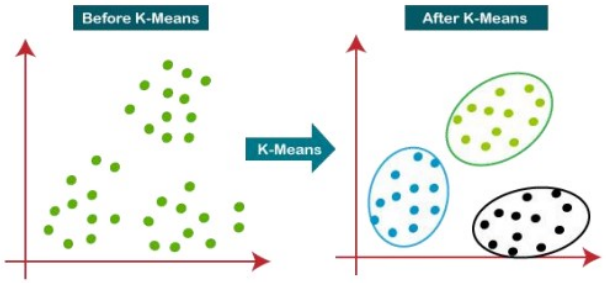

Fig. 2. K-means

A. Technique and Model

In this section,we will discuss the methodology that we have used. Initially, there is an auto-encoder model with two-tier architecture. For every different dataset, we separately configure model layers. Then the auto-encoder[2] is trained to minimize the reconstruction loss.

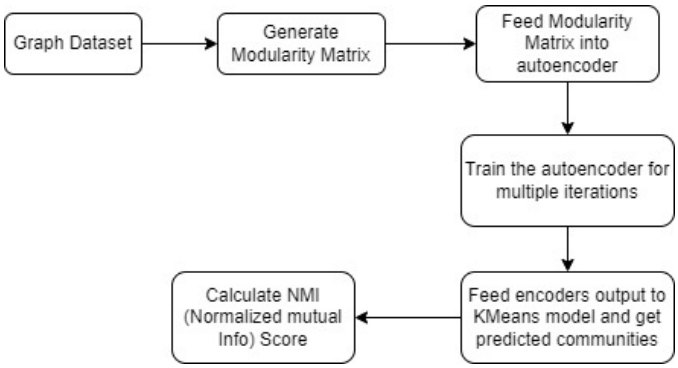

Fig. 3. Proposed Approach

Rather than directly giving an adjacency matrix as input we generate a modularity matrix[4,7] that would be feed as input. This modularity matrix is feed into our auto-encoder architecture[2,5,6]. It will learn the low-level depiction of our

input graph. As Now dimension is reduced we could use a clustering technique i.e K-means to detect communities in the input graph structure. The key motivation behind this approach is that after learning a low-level depiction of this graph dataset, it can be feed directly to an ML or clustering model. Applying K means to this low-level depiction will be much better than applying it to the whole graph as all the noises are removed and we are left with important latent information. So our model will learn better with this low-level depiction. We have chosen weight binding as an optimization technique whereby weights are distributed between tiers and the overall weight matrix will be optimized during the reverse propagation.

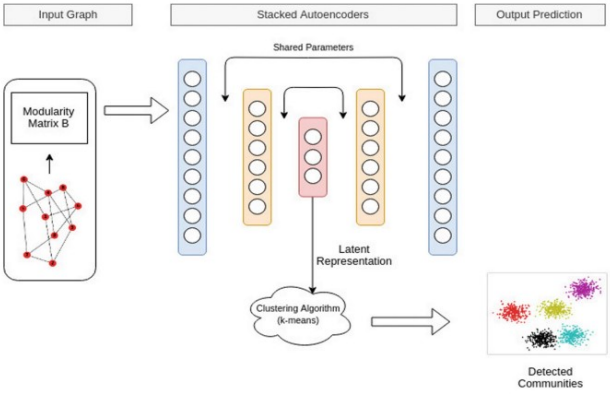

Fig. 4. Proposed Architecture

B. Datasets

The dataset that we are using is Dolphin Network, Polblogs Network, Polbooks Network and Football Network.

- **Dolphins Networks**: This dataset was developed by Lusseau et al. It is an undirected social network of frequent associations between 62 dolphins in a group residing off Doubtful Sound, New Zealand (2003)[13].
- **Polblog Networks**: It contains an undirected graphical network of blogs produced by editors during the 2005 US Presidential Election[1].
- **Polbooks Networks**: In this dataset, nodes depict Amazon.com's political books. As evidenced by the "customers who bought this book also bought these other books" feature on Amazon, edges depict frequent books purchased together by purchasers of the same book.[11].
- **Football Network**: This dataset contains 22 football teams that compete in the Paris World Championship(1998). Contracts with foreign countries are common among national team players. Players are exported from one country to another, therefore, creating an international market of players. Contracts were available in 35 countries for members of the 22 teams. A valued, asymmetric graph can be used to calculate the number of players being exported via team to different particular countries. The plot is not symmetrical because some countries just export and others only import [12].

C. Training

Custom Layers, the functionality of Keras is used to form neural network layers that will be capable of weight sharing. The optimizer that we are using is Adam and the activation function being used is relu. Sigmoid cross-entropy is used as a loss function. We also considered dropouts to be 20 per cent. We train our model for 100 epochs with a batch size equal to 16. Then we took the output from our encoder model in auto-encoder. We are using the K Means clustering technique to find clusters/communities in our graph. After that, we will train our model on football, dolphin, polbooks, and polblogs datasets.

VI. Result Analysis

We have decided to use NMI (Normalized Mutual Information) as the evaluation metric. It gives a score from 0 and 1, depending on the correlation between two different groups. Getting 0 means no common information, and 1 denotes perfect correlation. Mathematically, N.M.I is formulated as: Here:

$$NMI(Y,C) = \frac{2 \times I(Y;C)}{H(Y) + H(C)}$$

- Y : Class Label
- C : Clustering Label
- H() : Entropy
- I(Y;C) : Mutual information between Y and C.

We then analysed the output group which was produced by our proposed model with the ground truth value of the network/dataset.

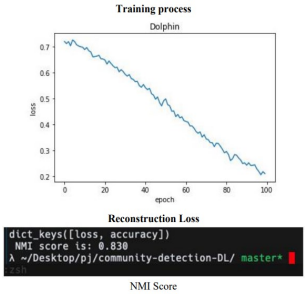

Fig. 5. Result analysis of Dolphins network

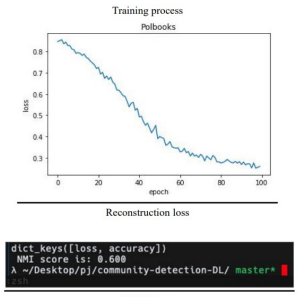

Fig. 6. Result analysis of Polbooks newtwork

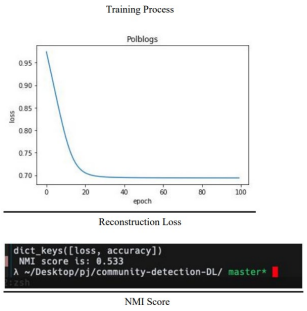

Fig. 7. Result analysis of Polblogs newtwork

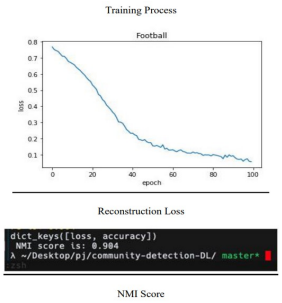

Fig. 8. Result analysis of Football newtwork

CONCLUSION

This document shows an improved method for the problem of community detection. For the unilateral networks, we used auto-encoders to minimise the reconstruction loss. Modularity matrix is feed as input which improved to analyse and it will learn low-level depiction of input-graph. Training on multiple datasets shows that the proposed method performs with better results than that of pre-existing state art of methods. The proposed method provides a simple framework for community discovery that allows the model to be applied to various sized networks with reduced training time.

REFERENCES

[1] Lada A. Adamic. The Political Blogosphere and the 2004 U.S. Election: Divided They Blog. Proceedings of the 3rd International Workshop on Link Discovery, 04 2005.

[2] Geoffrey E. Hinton and Richard S. Zemel. auto-encoders, Minimum Description Length and Helmholtz Free Energy. Advances in Neural Information Processing Systems, 6, 02 1994.

[3] Ioannis Psorakis, Stephen Roberts, Mark Ebden, and Ben Sheldon. Overlapping community detection using Bayesian nonnegative matrix factorization. Physical review. E, Statistical, nonlinear, and soft matter physics, 83:066114, 06 2011.

[4] Liang Yang, Xiaochun Cao, Dongxiao He, Chuan Wang, Xiao Wang, and Weixiong Zhang. Modularity based community detection with deep learning. 01 2016.

[5] Fanghua Ye, Chuan Chen, and Zibin Zheng. Deep Auto-Encoder like Nonnegative Matrix Factorization for Community Detection. In CIKM, 2018.

[6] Phi Vu Tran. Learning to Make Predictions on Graphs with auto-encoders. pages 237–245, 10 2018.

[7] Mark E.J. Newman. Modularity and community structure in networks. Proceedings of the National Academy of Sciences of the United States of America, 103:8577–82, 07 2006.

[8] Michelle Girvan and Mark E.J. Newman. Community structure in social and biological networks. proc natl acad sci, 99:7821–7826, 11 2001.

[9] David Lusseau and Micaleah Newman. Identifying the role that animals play in their social networks. Proceedings. Biological sciences, 271 Suppl 6:S477–81, 2004.

[10] P. Kumar, R. Jain, S. Chaudhary and S. Kumar, "Solving Community Detection in Social Networks: A comprehensive study," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 239-345, doi: 10.1109/ICCMC51019.2021.9418412.

[11] V. Krebs, Pol Books, unpublished, http://www.orgnet.com

[12] Dagstuhl seminar: Link Analysis and Visualization, Football network Dagstuhl 1-6. July 2001.

[13] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations, Behavioral Ecology and Sociobiology 54, 396-405 (2003).