ADSL Graph Hands-On 1
NetworkX & Karate Club

1. Hands-On Introduction

In this hands-on, you need to practice two graph package, NetworkX and Karate Club. The former will help you to load, store and study the graph structure, and the latter is an unsupervised machine learning library for NetworkX. After this hands-on, you will know how to manipulate and analyze your graph data, then perform unsupervised graph embedding on it. Please read the following content for more detail.

2. NetworkX

A. Introduction

NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. It provides

- tools for the study of the structure and dynamics of social, biological, and infrastructure networks;
- a standard programming interface and graph implementation that is suitable for many applications;
- a rapid development environment for collaborative, multidisciplinary projects;
- an interface to existing numerical algorithms and code written in C, C++, and FORTRAN; and
- the ability to painlessly work with large nonstandard data sets.

With NetworkX you can load and store networks in standard and nonstandard data formats, generate many types of random and classic networks, analyze network structure, build network models, design new network algorithms, draw networks, and much more.

The following section will mainly follow the content in NetworkX Document. You can also refer to the document to learn more complete knowledge.

Before starting, please install NetworkX by the official document. Make sure you can import networkx in your environment. Besides, if you want to plot the graph, please also install “matplotlib”.

B. Graph Types

There are mainly four types of graph in NetworkX: Graph, DiGraph, MultiGraph and MultiDiGraph. The difference between them is as follows

Networkx Class	Type	Self-loops allowed	Parallel edges allowed
Graph	undirected	Yes	No
DiGraph	directed	Yes	No

MultiGraph	undirected	Yes	Yes
MultiDiGraph	directed	Yes	Yes

We will mainly talk about the **Graph Class,** while others are much similar with it. You can also refer to the document for other class.

C. Basic usage tutorial

Please refer to "networkx_practice.ipynb" to see the usage of networkx, or you can also read the official tutorial.

D. Tasks

In this section, please fulfill the requirements and get the answer.

a. Create a graph, which has five nodes and can be formed as a ring (with edges (1,2), (2,3), (3,4), (4,5), (5,1)).
Find the **average clustering coefficient** of this graph, and **calculate the value by hand** to check whether the value is correct or not.

b. Create a complete graph with 5 nodes by graph generator.
Find the **average degree** and **plot the graph with no node label**.

c. Load our graph data "`elliptic_txs_edgelist.csv`", and do the following things.
   1. Model the graph as a directed graph. The first entry is the start node, the second is the end node.
   2. Get the number of nodes
   3. Get the number of edges
   4. Get the number of weakly connected component
   5. Plot the top three smallest component according to the number of nodes.

3. Karate Club

A. Introduction

Karate Club is an unsupervised machine learning extension library for NetworkX. It builds on other open source linear algebra, machine learning, and graph signal processing libraries such as Numpy, Scipy, Gensim, PyGSP, and Scikit-Learn. *Karate Club* consists of state-of-the-art methods to do unsupervised learning on graph structured data. To put it simply, it is a Swiss Army knife for small-scale graph mining research. First, it provides network embedding techniques at the node and graph level. Second, it includes a variety of overlapping and non-overlapping commmunity detection methods. Implemented methods cover a wide range of network science (NetSci, Complenet), data mining (ICDM, CIKM, KDD), artificial intelligence (AAAI, IJCAI) and machine learning (NeurIPS, ICML, ICLR) conferences, workshops, and pieces from prominent journals.

B. Basic usage tutorial

The usage of Karate is very simple, please read the overview in their official website.

C. Task - Node Classification

In this task, you need to generate the node embedding by unsupervised graph learning module, and evaluate the embedding quality by node classification task with Logistic Regression classifier.

1. Loading dataset "**facebook**" by **karateclub.dataset.GraphReader**.
2. Get the node embedding of the graph by any kinds of the **unsupervised node embedding** provided in karateclub.
3. To evaluate the quality of the node embedding, you need to take a downstream task and see the performance of this task.
   Here we'll use **Logistic Regression** in Scikit-Learn to train and test the node label.
4. Notice that you can use **train_test_split** or **cross_validate** or any kind of data splitting function you like. Take it easy to practice these modules and try to make the performance better!