Lab 03: Community Detection

Instructor: Doan Dinh Toan
Department of Computer Science,
Faculty of Information Technology
University of Science - VNU-HCM
toandd.i81@gmail.com

Abstract

In this assignment, you will work on Girvan-Newman algorithm for detecting communities in social networks. This assignment aims to help you understand the community detection problem and how to handle it with widely available graph processing tools. [1]

0 Preliminary

0.1 Reminder

The main objective of this course is to learn and truly learn. You can discuss this with your classmate, but you need to take responsibility for your submission, which actually depends on your understanding of this course. For any kind of cheating and plagiarism, students will be graded 0 marks for the whole course.

0.2 Submission guideline

Each team submits its result to a folder named `teamABC`, with `ABC` being the team's name. The folder structure is as follows:

```
teamABC
├── src
├── docs
│   ├── report.md
│   ├── report.pdf
│   └── images
└── readme.md
```

- `src` is the folder for your source code.
- `docs` is the folder for your documents, including the work report and images associated with your report. If the lab assignment requires screenshots as proof, the images need to be stored in this folder if you inserted them in the report.
  - `report.md` is your report file in MarkDown format. The report must be written in English. This assignment will come with a template folder that already has a report template (you can use my OSCP template or create your own). If you are not familiar with MarkDown, see this cheat sheet. The report must include the following items:
    * Your team's result (How much work, in percent (%), have you finished in each section?)
    * The answer to each section's tasks.

[1] This assignment is created based on an INF553 Foundations and Applications of Data Mining course assignment. The following paragraphs are from the webpage with some modifications.

    - Reflection of your team. (Does your journey to the deadline have any bugs? How have you overcome it? What have you learned after this process? If you cannot overcome the bugs, describe where the bottlenecks are in your work.
    - References to your work.
  - `report.pdf` is the PDF file of your report, converted from the MarkDown file mentioned above.
- `readme.md` is the file that introduces your team and this lab assignment, this file should include the following basic information:
  1. Information about the course, the assignment, and notes to the instructors (if any).
  2. Information about your team (Student ID, full name of each member).

0.3 Rubrics

- Task 01 (60%)
- Task 02 (30%)
- Quality of the report (10%)

0.4 Notes

- If you complete the project using Google Colab, please provide the Google Colab link(s). The data should be organized such that the grader can run your code with minor adjustments. No modification after the deadline is allowed.
- Alternatively, please submit the whole source code (in Python only) and write a careful guide of how to run the code.

1 From raw data to graph

You aim to find users with similar business tastes from the dataset given in the CSV file, ub_sample_data.csv.

You need to construct the social network from the given data. Each node represents a user, and there will be an undirected edge between two nodes if the number of times that two corresponding users review the same business is greater than or equivalent to some predefined filter threshold. For example, suppose user1 reviewed [business1, business2, business3] and user2 reviewed [business2, business3, business4, business5]. Then, if the threshold is 2, there will be an edge between user1 and user2.

If the user node has no edge, we will not include that node in the graph.

In this assignment, we use filter threshold 7.

Please use user_id directly as strings when constructing the graph, don't hash them to integers.

2 Task 1: Community Detection using Girvan-Newman algorithm (GM)

In this task, you first construct the graph following the specification given in Section 1 and detect the communities in the constructed graph by using Girvan-Newman algorithm.

GM [1] detects communities by progressively removing edges with high betweenness scores from the original network; the connected components of the remaining network are the communities. For the algorithm details, you may also refer to the Mining of Massive Datasets book – Chapter 10.

You need to implement the graph construction and the Girvan-Newman algorithm by yourself. Pandas DataFrame is possible.

2.1 Betweenness Calculation

You calculate the betweenness score for every edge in the constructed graph and then save the result in a text file whose format is as follows.

('user_id1', 'user_id2'), betweenness value

The two user_ids in each tuple must be in lexicographical order. You first sort the betweenness values in descending order. Then, you sort the first user_ids (of string type) in all the tuples in lexicographical order. You do not need to round your result.

2.2 Community Detection

You divide the graph into an appropriate number of communities to reach the global highest modularity. The formula of modularity is shown below.

$$\mathcal{Q} = \sum_{s \in S} (\texttt{(\#edge with in group s)} - \texttt{(expected \#edges with in group s)}) \quad (1)$$

$$\mathcal{Q}(G, S) = \frac{1}{2m} \sum_{s \in S} \sum_{i \in s} \sum_{j \in s} (A_{ij} - \frac{k_i k_j}{2m}) \quad (2)$$

with normalizing cost: $-1 < Q < 1$ and $A\_ij = 1$ if i connects j, otherwise $A\_ij = 0$.

The GM algorithm requires recomputing the betweenness scores after removing one edge. In the formula, "m" stands for the number of edges in the original graph, and "A" represents the adjacent matrix of the original graph.*(Hint: in each removal step, we do not alter "m" and "A").*

If the community only has one user node, we still regard it as a valid community.

You need to save the resulting communities in a text file. Each line represents one community in the following format.

'user_id1', 'user_id2', 'user_id3', 'user_id4', ...

The user_ids in each community must be in lexicographical order. Your first sort the communities ascendingly by their sizes. Then, you sort all the first user_ids (of string type) in the communities in lexicographical order.

The last line shows the global highest modularity.

```
'111', '681'
'1231', '142'
'2281', '283'
'2517', '2744'
'2862', '2985'
'359', '468'
'659', '661'
'102', '125', '54'
'166', '245', '58'
'119', '1615', '2543', '8'
'2', '216', '35', '6'
'1530', '1992', '2116', '497', '935'
'120', '183', '209', '60', '728', '74'
'1245', '1794', '1866', '2113', '2150', '2188', '2606', '2876', '2953', '2955', '640'
'1072', '1270', '1565', '1620', '1761', '1861', '2479', '2575', '2976', '30', '3280', '475', '713', '752'
'1136', '1197', '1206', '1355', '1408', '1418', '1498', '1508', '1648', '1723', '1913', '1918', '2005', '2097', '2332', '23
```

Figure 1: community output file format

3 Task 2: Community Detection using Girvan-Newman algorithm in NetworkX

In this task, you repeat what required in Task 1, i.e., to detect the communities in the given graph by GM algorithm such that a maximum modularity is obtained. Then, you compare the results provided by NetworkX GM and your GM implementation.

You need to use NetworkX API to construct the graph and run the Girvan-Newman algorithm.

The following websites may help you get started with the NetworkX.

References

[1] M Girvan and M E J Newman. Community structure in social and biological networks. *Proc Natl Acad Sci U S A*, 99(12):7821–7826, June 2002.