Social Network Analysis
Final Project

2019

Introduction

The Final Project composes of three assignments: Data Collection, Analytical Tasks, Report. Each part has a dedicated GitHub repository.

General Rules

1. Groups should be composed of, at most, 4 students;
2. All students need to have a GitHub (`https://github.com/`) account;
3. Every group must access and register to the GitHub classroom repositories provided;
4. Code, data, and report must be uploaded (pushed) on such repositories;
5. The final version of the report must be submitted (in pdf) through SNA Course Moodle (1 submission per group).

Before starting the project, send an email to the course instructor specifying: name surname and student id of all group members (along with the planned data source).

1 Assignment 0: Data Collection

Access your copy of the data collection repository[1].

Data collection can be carried out without any restriction on programming languages (python is only a warm suggestion) and/or online sources. The adoption of already crawled network datasets available on the internet (e.g., networks from networkrepository, socialcomputing, snap, konect...) will not be considered as a valid option.

[1] `https://classroom.github.com/g/SqreoZAe`

1.1 Workflow

Identify an online data source, Crawl data from it (or collect them through API if available), Build up a network from the data!

1.1.1 Requirement

- The network must have at least 10-15k nodes. Specific cases involving the analysis of smaller networks must be discussed beforehand with the instructor.
- The produced code must be stored into the code folder: please, briefly comment the choices made/strategies adopted to perform the crawling.
- The final version of the data (i.e., the network and, if present, all additional data) must be compressed and stored into the data folder.

1.1.2 Data Sources ideas

Twitter, Last.fm, Blogs, Reddit, Blabla car, Linkedin, Corpora, Wikipedia, Newspaper...

2 Assignment 1: Network Analysis and Analytical Tasks

Access your copy of the analytical tasks repository[2].

This assignment composes of 4 exercises.

2.1 Exercise 0: Network Analysis

The analysis can be performed either by using a visual tool (i.e., Cytoscape and Gephi) and/or the support of a programming language. The use of python (networkx or igraph) is not mandatory, although, strongly suggested.

Network analysis must include at least:

- Degree distribution analysis;
- Connected components analysis;
- Path analysis;
- Clustering Coefficient, Density analysis;
- Centrality analysis.

Moreover, the statistics computed on the crawled data must be compared with the ones of (i) random and (ii) preferential attachment graphs having the same number of nodes and edges.

[2] `https://classroom.github.com/g/ip0jk6Xl`

2.2 Exercise 1-2: Analytical Tasks

Each group must address *at least* two among the following tasks:

Community Discovery: Identify, evaluate and validate the modular structure of the crawled network sample. The results of K-clique, Label Propagation, Louvain, and Demon/Angel must be evaluated and compared. If additional semantic information for the analysed graph are present use them to make sense of the identified partitions. For CD algorithm implementations (as well as for their evaluation and comparison) refer to the cdlib library. The analysis can be extended selecting approaches considered interesting among the one present in such library.

Community Discovery 2: Define, implement and test either an existing or a novel Community Discovery approach not yet present in CDlib. For a list of well-known approaches refer to the Fortunato and Coscia's surveys.

Spreading: Simulate, using the ndlib python library, the diffusion models discussed during the course (i.e., SI, SIS, SIR and Threshold model) both on the crawled data and on synthetic graphs (i.e., ER and BA). Analyse the simulation results varying both model parameters and initial conditions (i.e., the infection seeds);

Spreading 2: Leveraging the Custom Model facility offered by ndlib design an ad-hoc, novel, diffusion model for the crawled graph. The model can be designed to take advantage of both network topological characteristics as well as external semantic information attached to nodes/edges (if present). Define your model so to solve a specific diffusion problem you consider interesting for your data. The model can be either coded in python or expressed using NDQL. Analyse the results varying both model parameters and initial conditions (i.e., the infection seeds);

Link Prediction: Partition each network in a training (80% of the edges) and a test set (20% of the edges) and apply some of the classical unsupervised link prediction approaches introduced in "David Liben-Nowell, Jon M. Kleinberg: The link prediction problem for social networks. CIKM 2003" (i.e. Common Neighbors, Adamic Adar, Jaccard, Preferential Attachment). Discuss the prediction accuracy as done in the referenced paper.

Link Prediction 2: Following the same rationale of the previous exercise, build up a supervised approach[3] to link prediction using a classifier. Define the features, test the model(s), evaluate and discuss the results.

Network Resilience: Define a set of measures to compute tie strength and analyze the impact of strong/weak ties on the connectedness and resilience of the crawled network.

[3]This exercise requires knowledge of Data Mining tools and techniques.

Graphlets: Graphlets are small, connected, non-isomorphic[4] induced subgraphs[5] of a large network. The size of a graphlet is the number of the nodes it is composed of: for a same size multiple graphlets may exist[6]. Define an approximate algorithm that allows to estimate the number of graphlets of size 3 and 4 and test it on your data. Which are the most frequent graphlets?

2.3 Exercise 3: Open problem

Define a research question on your data and use SNA tools to address it!

This final task requires you to:

- reason on the crawled data,
- identify a concrete question to address, and
- try to tackle it combining the technique discussed in class/implementing your own ideas/solutions.

If necessarily you can make use of additional data w.r.t. the network structure.

3 Assignment 2: Project Report

Access your copy of the report repository[7].

Discuss the result of all the analysis (assignments 0 and 1) in a written report:

- Specify group members and link the GitHub repositories in the first page;
- Focus on the analytical methodologies and obtained results (it is mandatory to provide an interpretation of the analysis outcome);
- Max 12 pages, double column (use the template in the report folder).

[4]Graph isomorphism is an equivalence relation on graphs: two graphs G and H are said to be isomorphic if there exist a function f such that any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. This kind of bijection is commonly described as "edge-preserving bijection".

[5]An induced subgraph must contain all edges between its nodes that are present in the original network.

[6]A single graphlet exists only among 2 nodes. Considering all the possible ways to connect 3 nodes, two different graphlets can be identified: the chain, the triangle.

[7]`https://classroom.github.com/g/W1ue58Ys`