Project proposal

Alberto Rigon, Luca Scattolaro, Alessandro Chimetto

23 October 2021

The goal of this project consists in finding significant and important nodes in a social network of Twitch users by computing node features and clustering coefficient and by finding the motifs of the network.

1 Motivation

Social network analysis is a field of study that attempt to understand relationships between entities in a network based on the assumption about the “importance” of relationship between entities. Twitch and Social Live-Streaming (SLS) services are recent internet phenomena that support massive amounts of users congregating together around common interests to form interactive and social communities.

- We want to find the most influent people in this Social Network since it is an important task that may be useful for business, marketing and statistical purposes. We will do this by computing several centrality and clustering coefficient statistics.
- We want to understand what motifs are more surprising in Twitch in order to discover and analyze the structure of the Twitch network and see what we can understand from it.

1.1 Data

We are going to use a social network of Twitch users collected from the public API in Spring 2018. Nodes are Twitch users and edges are mutual follower relationships between them. The graph forms a single strongly connected component without missing attributes. The dataset can be found here.

Here we report some dataset statistics:

Property	Value
Directed	No
Node features	No
Edge features	No
Node labels	No
Temporal	No
Nodes	168,114
Edges	6,797,557
Density	0.0005
Transitivity	0.0184

1.2 Method

- Problem: compute the centrality scores for all nodes using exact algorithms and also their approximations in order to compare the computational time and error:
  - Closeness centrality
    - Exact version
    - Eppstein-Wang algorithm
  - Betweeness centrality
    - Exact version
    - Approximate version
- Problem: compute the clustering coefficient for the graph:

  - Local clustering coefficient
  - Global clustering coefficient

- Problem: compute the motifs of the network:
  - Algorithms: the ones that we will see during the course (since we did not cover this topic yet)

2 Intended experiments

2.1 Implementations

We will use the following implementations:

- Closeness centrality
  - Exact version: `https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.centrality.closeness_centrality.html#networkx.algorithms.centrality.closeness_centrality`
  - Eppstein-Wang algorithm: we will implement it ourselves (using the pseudocode we have seen in the lessons)
- Betweeness centrality
  - Exact version: `https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.centrality.betweenness_centrality.html#networkx.algorithms.centrality.betweenness_centrality`
  - Approximate version: we will implement it ourselves (using the pseudocode we have seen in the lessons)
- Clustering coefficient
  - Local clustering coefficient: we will implement it ourselves (using the pseudocode we have seen in the lessons)
  - Global clustering coefficient: we will implement it ourselves (using the pseudocode we have seen in the lessons)
- Motifs: we don't know yet, we will choose after we have covered this topic in class

2.2 Machines

- PC1
  - CPU: Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz
  - RAM: 6,0 GB
- PC2
  - CPU: Intel(R) Core(TM) i5-3570K CPU @ 3.40GHz
  - RAM: 8,0 GB
- PC1
  - CPU: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz-1.80 GHz
  - RAM: 16,0 GB
- PC4
  - CPU: 11th Gen Intel(R) Core(TM) i5-11600K @ 3.90GHz
  - RAM: 32,0 GB

2.3 Experiments

- Compute centralities on the data, compare the results and create the rankings
- Compute clustering coefficients and analyze them
- Compute motifs and try to understand what these motifs represent