Comparison of Network Analytics and Significance Analysis on Spotify Artist Feature Collaboration Network

Learning From Networks - Project proposal

Fabio Cociancich, Luca Fantin, Alessandro Lincetto

Master Degree in Computer Engineering - University of Padova

I. Motivation

Nowadays, Spotify is one of the most popular music streaming services in the world. As such, analyzing its usage data can reveal interesting information about music trends. In this project, we focus our analysis on the detection of the most popular artists, see how its results change when considering only certain genres and popularity levels, determine how closely connected the artists are and whether our findings can be considered interesting.

II. Dataset

We will consider a graph where nodes correspond to artists and edges connect artists who have collaborated on at least one song. In particular, the dataset we intend to use for this project is the Spotify Artist Feature Collaboration Network from Kaggle [1]. This graph has 156.422 nodes, which include around 20,000 artists who appear in the Spotify weekly charts and around 136,000 artists who had at least one feature with the chart artists, and 300,387 edges between them. The information included with the nodes allow for the analyses suggested above:

- number of followers, according to the Spotify API
- artist popularity, expressed as an integer number between 0 and 100 (100 corresponding to the most popular artist on the service), according to the Spotify API
- list of genres, according to the Spotify API
- list showing the number of Spotify chart hits in different countries, according to the data collected by kworb.net

III. Method

To achieve out targets, our work is divided in two phases. The first one consists of the computation of several network analytics. On one hand, we will compare the centrality measures presented during the lectures (closeness, betweenness, PageRank) with *eigenvector centrality*. The latter is built on the intuition that a node is important if it is connected to other important nodes. Given a graph $G = (V, E)$, let us define $\mathbf{x} \in \mathbb{R}^{|V|}$ the vector of the centrality values for all nodes in G, A the adjacency matrix of G and $\lambda \neq 0$ a constant. For any node i we can write:

$$x_i = \frac{1}{\lambda} \sum_{j=1}^{|V|} A_{i,j} x_j \quad \rightarrow \quad Ax = \lambda x$$

The mathematical representation of the intuition can thus be reformulated as finding the eigenvector of the adjacency matrix corresponding to the eigenvalue λ; such vector includes the values of the eigenvector centrality for all nodes. This centrality measure has been studied extensively [2] [3] [4], also in the context of social media network analysis [5], including Spotify [6] [7]. Furthermore, we want to compute these measures on subgraphs of the whole network, considering only certain music genres and popularity threshold.

A similar analysis will be done for the clustering coefficients. We will compute its values for the whole graph and for the aforementioned subgraphs.

The second part of our work will consist of statistical hypothesis testing on the node analytics computed on our graph against those computed on a series of random graphs. We intend to generate them with the *Holme-Kim algorithm* [8]. This model resembles real-world networks, such as social networks, more closely than the traditional Erdős-Rényi model. Instead of assuming a fixed number of nodes and uniform distribution for the probabilities of adding new edges, this new model starts from a certain number of nodes, iteratively adds new nodes and connects them with already existing nodes with a distribution that favours nodes with an already high degree. This allows the resulting graph to achieve a power-law distribution of the degrees: as we consider higher degrees, the number of nodes with such degree decreases exponentially. Such characteristic is observed in many real-world networks, thus we expect it to arise in our Spotify artists' graph as well. Furthermore, this model produces graphs with higher, tunable clustering coefficients by creating additional edges: once a newly created node v is connected to an existing one w, a new edge is created between v and one of the neighbours of w.

Further investigation will be needed to determine if and how we need to fix node features such as genre and popularity level when generating the random graphs.

IV. IMPLEMENTATION

The programs needed for our analyses will be implemented in Python. For the centrality measures we will use the implementations available at NetworkX package [9] which contains methods for computing various centrality measures such as closeness, betweenness, eigenvector and PageRank.

For the approximate algorithms we will use the NetworKit package [10]. Some methods are available, such as the approximate Checkin-Cohen-Kaplan algorithm for the closeness centrality [11] and the approximation algorithm for betweenness centrality developed by Riondato and Kornaropoulos [12]. Another betweenness centrality approximate algorithm is SILVAN, developed by Leonardo Pellegrina and Fabio Vandin. [13]

The NetworKit package will also be used for evaluating the global clustering coefficients, with both exact and approximate algorithms. [14]

The NetworkX package will also be used for the graph generators [15]. There is a specified method for generating models using the Holme-Kim algorithm.

V. MACHINES

Our programs will be executed on our local machines. They all employ AMD Ryzen 5/7 CPUs and RAMs ranging from 8 GBs and 24 GBs. We will also investigate the possibility to execute them on the CAPRI High-Performance Computing (HPC) system [16]. This system features the following hardware:

- 16 Intel(R) Xeon(R) Gold 6130 @ 2.10GHz CPUs
- 6 TB DDR4 RAM
- 2 NVIDIA Tesla P100 16GB GPUs
- 40 TB of disk space

CONTRIBUTIONS

For this proposal, Fabio Cociancich and Alessandro Lincetto researched and conducted the first tests with the Python packages, and wrote the related "Implementation" section. This can be quantified as 30% of the work for each of them. Luca Fantin wrote instead the rest of the proposal and researched the dataset, the eigenvector centrality measure and the Holme-Kim random graph model. This can be quantified as 40% of the work.

REFERENCES

[1] Julian Freyberg. *Spotify Artist Feature Collaboration Network*. https://www.kaggle.com/datasets/jfreyberg/spotify-artist-feature-collaboration-network.

[2] Phillip Bonacich. “Some unique properties of eigenvector centrality”. In: *Social Networks* 29.4 (2007), pp. 555–564. ISSN: 0378-8733. DOI: https://doi.org/10.1016/j.socnet.2007.04.002.

[3] Stephen Borgatti, Kathleen Carley, and David Krackhardt. “On the Robustness of Centrality Measures Under Conditions of Imperfect Data”. In: *Social Networks* 28 (May 2006), pp. 124–136. DOI: 10.1016/j.socnet.2005.05.001.

[4] Leo Spizzirri. "Justification and application of eigenvector centrality". In: *Algebra in Geography: Eigenvectors of Network* (2011). URL: https://sites.math.washington.edu/~morrow/336_11/papers/leo.pdf.

[5] Warih Maharani, Adiwijaya, and Alfian Akbar Gozali. "Degree centrality and eigenvector centrality in twitter". In: *2014 8th International Conference on Telecommunication Systems Services and Applications (TSSA)*. 2014, pp. 1–5. DOI: 10.1109/TSSA.2014.7065911.

[6] Tobin South, Matthew Roughan, and Lewis Mitchell. "Popularity and centrality in Spotify networks: critical transitions in eigenvector centrality". In: *Journal of Complex Networks* 8.6 (Mar. 2021), cnaa050. ISSN: 2051-1329. DOI: 10.1093/comnet/cnaa050. URL: https://doi.org/10.1093/comnet/cnaa050.

[7] Tobin South. "Network analysis of the Spotify artist collaboration graph". In: *Australian Mathematical Sciences Institute* (2018), pp. 1–12. URL: https://vrs.amsi.org.au/wp-content/uploads/sites/84/2018/04/tobin_south_vrs-report.pdf.

[8] Petter Holme and Beom Jun Kim. "Growing scale-free networks with tunable clustering". In: *Physical Review E* 65.2 (Jan. 2002). ISSN: 1095-3787. DOI: 10.1103/physreve.65.026107. URL: http://dx.doi.org/10.1103/PhysRevE.65.026107.

[9] *NetworkX, centrality algorithms.* URL: https://networkx.org/documentation/stable/reference/algorithms/centrality.html.

[10] *NetworKit, centrality.* URL: https://networkit.github.io/dev-docs/python_api/centrality.html#module-networkit.centrality.

[11] Shiri Chechik, Edith Cohen, and Haim Kaplan. *Average Distance Queries through Weighted Samples in Graphs and Metric Spaces: High Scalability with Tight Statistical Guarantees.* 2015. arXiv: 1503.08528 [cs.SI]. URL: https://arxiv.org/abs/1503.08528.

[12] Matteo Riondato and Evgenios M. Kornaropoulos. "Fast approximation of betweenness centrality through sampling". In: *Data Mining and Knowledge Discovery* 30.2 (Mar. 2016), pp. 438–475. ISSN: 1573-756X. DOI: 10.1007/s10618-015-0423-0. URL: https://doi.org/10.1007/s10618-015-0423-0.

[13] Leonardo Pellegrina and Fabio Vandin. *SILVAN: Estimating Betweenness Centralities with Progressive Sampling and Non-uniform Rademacher Bounds.* 2022. arXiv: 2106.03462 [cs.DS]. URL: https://arxiv.org/abs/2106.03462.

[14] *NetworKit, global clustering coefficient.* URL: https://networkit.github.io/dev-docs/python_api/globals.html#networkit.globals.ClusteringCoefficient.

[15] *NetworkX, graph generators.* URL: https://networkx.org/documentation/stable/reference/generators.html.

[16] University of Padova Strategic Research Infrastructure Grant 2017. *CAPRI: Calcolo ad Alte Prestazioni per la Ricerca e l'Innovazione.* https://capri.dei.unipd.it/.