

Spotify Artists Network Analysis

Massimo Rondelli

massimo.rondelli@studio.unibo.it

Department of Computer Science and Engineering,
University of Bologna

Nadia Farokhpay

nadia.farokhpay@studio.unibo.it

Department of Computer Science and Engineering,
University of Bologna

Michele Dinelli

michele.dinelli5@studio.unibo.it

Department of Computer Science and Engineering,
University of Bologna

Youssef Hanna

youssefawni.hanna@studio.unibo.it

Department of Computer Science and Engineering,
University of Bologna

1 Introduction

Music taste is a sort of a fingerprint that identifies each of us, of course there are people who care more or less about music, but everyone must have a peculiar musical taste that is unique. People tend to group their favorite music in structures the so-called music playlist, list of songs where genres, artists and even musical periods are mixed together forming the aforementioned fingerprint, that belongs to each of us. When we thought about music playlists we realized that there must exist some sort of relationship between artists that share place inside them. This report analyzes Spotify [1] music playlists, in particular the relationship between artists that appear in the same playlists.

2 Problem and Motivation

The rise of digital music platforms such as Spotify has impacted not only how people consume music, but also how they express their musical tastes and identity through personal playlists. While each playlist is a unique expression, when combined, they reveal patterns. Certain artists may be more likely to appear together, implying latent associations based on genre, mood, popularity, or user behavior. Understanding these connections between artists is not a simple task. Unlike traditional music recommendation systems that rely on metadata or user ratings, co-occurrence data implicitly embedded in playlists captures real-world, user-curated musical context. The motivation behind this analysis is to uncover the hidden structure within user-generated playlists by building and analyzing a co-occurrence network of artists.

3 Dataset

The dataset used is the continuation of the RecSys Challenge 2018 [2] and it contains 1,000,000 music playlists, created by users on the Spotify platform between January 2010 and October 2017. The dataset's original task is automatic playlist continuation: given a seed playlist title and/or an initial set of tracks in a playlist, predict the next tracks. The dataset is publicly available [3] but it's not user-friendly because i) it's divided in multiple slices formatted as json files, ii) it's huge (5GB). We randomly sampled 5 dataset slices in order to drastically reduce the dimension of the upcoming network. This left us with 5000 playlists. Since we cannot (computationally) analyze such a network we computed a subgraph i) by reducing the number of playlist ii) by keeping connected components only.

The result is a monomodal, undirected and weighted co-occurrence network. Nodes represent artists, edges represent the co-occurrence of artists in the same playlist and edge weights are the product of the artists' frequencies in that playlist.

4 Validity and Reliability (lorem)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opine-mur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae sine metu degendae praesidia firmissima. – Filium morte multavit. – Si sine causa, nollem me ab eo delectari, quod ista Platonis, Aristoteli, Theophrasti orationis ornamenta neglexerit. Nam illud quidem physici, credere aliquid esse minimum, quod profecto numquam putavisset, si a Polyano, familiari suo, geometrica discere maluisset quam illum etiam ipsum dedocere. Sol Democrito magnus videtur, quippe homini erudito in geometriaque perfecto, huic pedalis fortasse; tantum enim esse omnino in nostris poetis aut inertissimae segnitiae est aut fastidii delicatissimi. Mihi quidem videtur, inermis ac nudus est. Tollit definitiones, nihil de dividendo ac partiendo docet, non quo ignorare vos arbitrer, sed ut.

5 Measures and Results

Computing network measures helps one to summarize and extract insight from complex network data in a manageable and interpretable way. Analyzing the raw network directly is impractical due to its size and complexity. Music playlists, like many real-world networks, form intricate webs of relationships. By applying mathematical measures that capture interesting features of network structure quantitatively it's easier to extract important information such as structural roles, patterns and node importances [4].

5.1 Centrality Measures

Network centrality metrics quantify the influence or importance of a node in a network however, there is no generalized definition of centrality [5], as a results many centrality measures exists [6].

5.1.1 Degree Centrality

The most simple yet illuminating centrality metric is degree centrality. In an undirected graph it's defined by the number of edges attached to a node. Defining this in terms of the network's adjacency matrix A with N nodes gives

$$c_i = \sum_{j=1}^N A_{ij} \quad (1)$$

Artists with a high degree of centrality appear alongside many different artists in playlists, that means they are broadly connected across listener tastes (likely popular). Artists with a high co-occurrence rate are more likely to be frequent collaborators or playlist “glue” (these artists may be useful for playlist generation or recommendation models).

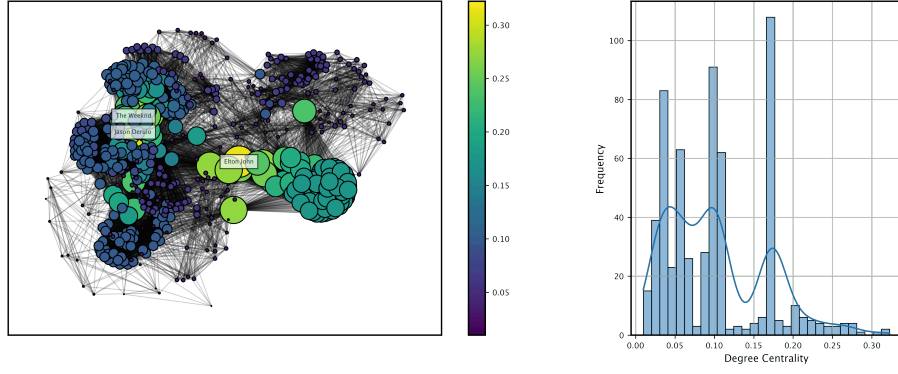


Figure 1: Visualization of degree centrality in the artist co-occurrence network. On the left: network graph where each node represents an artist and edges indicate co-occurrence in playlists. Node color and size are proportional to the node’s degree centrality, with brighter and larger nodes having higher centrality. The top three most central artists are Elton John, Jason Derulo, and The Weeknd. On the right: distribution of degree centrality values across the network. The histogram shows a heavy-tailed distribution, indicating that while most artists have low centrality, a few act as major hubs.

5.2 Weighted Degree Centrality

Weighted degree centrality is an extension of degree centrality metrics that takes into account edge weights in the network, which in this analysis represent the strength of artist co-occurrence, i.e., how often two artists appear together in playlists. In a weighted, undirected graph, weighted degree centrality measures the sum of the weights of edges incident to a node, capturing not only how many connections a node has but also how strong those connections are. Given a weighted adjacency matrix $W \in R^{N \times N}$, where W_{ij} denotes the weight of the edge between nodes i and j , the weighted degree centrality $c_{i(w)}$ of node i is defined as:

$$c_i^{(w)} = \sum_{j=1}^N W_{ij} \quad (2)$$

The superscript (w) just indicates “weighted”.

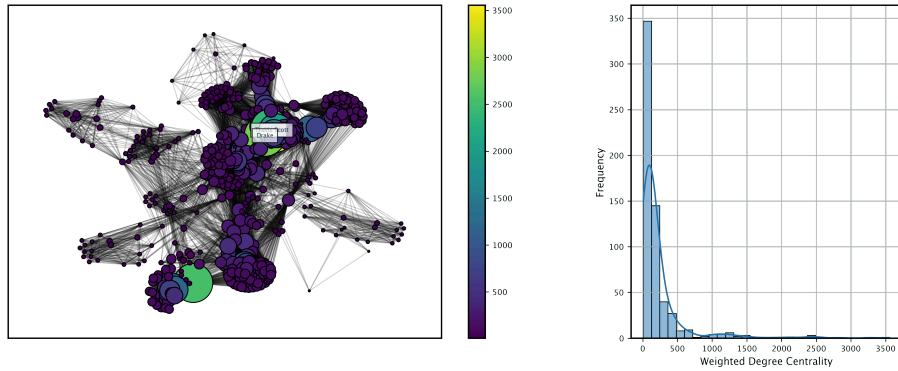


Figure 2: Visualization of weighted degree centrality in the artist co-occurrence network. On the left: network graph where node size and color represent the weighted degree centrality, accounting for the frequency of artist co-occurrence. Nodes with higher total edge weights appear larger and brighter. The most central artists by this metric are Drake, Travis Scott, and Ed Sheeran. On the right: histogram of weighted degree centrality values, showing a highly skewed distribution. This suggests that a few artists dominate playlist co-occurrences, likely due to frequent collaborations or broad popularity.

5.2.1 Eigencentrality

Centrality can be recursively defined in terms of the centrality of a node's neighbourhood. This comes from the notion that a node's importance in a network is increased by having connections to other nodes that are themselves important.

6 Conclusion

7 Critique

Figures

- Figure 1 Visualization of degree centrality in the artist co-occurrence network. On the left: network graph where each node represents an artist and edges indicate co-occurrence in playlists. Node color and size are proportional to the node's degree centrality, with brighter and larger nodes having higher centrality. The top three most central artists are Elton John, Jason Derulo, and The Weekend. On the right: distribution of degree centrality values across the network. The histogram shows a heavy-tailed distribution, indicating that while most artists have low centrality, a few act as major hubs. 3
- Figure 2 Visualization of weighted degree centrality in the artist co-occurrence network. On the left: network graph where node size and color represent the weighted degree centrality, accounting for the frequency of artist co-occurrence. Nodes with higher total edge weights appear larger and brighter. The most central artists by this metric are Drake, Travis Scott, and Ed Sheeran. On the right: histogram of weighted degree centrality values, showing a highly skewed distribution. This suggests that a few artists dominate playlist co-occurrences, likely due to frequent collaborations or broad popularity. 3

Bibliography

- [1] Wikipedia, "Spotify — Wikipedia, L'enciclopedia libera." [Online]. Available: <http://it.wikipedia.org/w/index.php?title=Spotify&oldid=144443141>
- [2] C.-W. Chen, P. Lamere, M. Schedl, and H. Zamani, "Recsys challenge 2018: automatic music playlist continuation," in *Proceedings of the 12th ACM Conference on Recommender Systems*, in RecSys '18. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2018, pp. 527–528. doi: [10.1145/3240323.3240342](https://doi.org/10.1145/3240323.3240342).
- [3] Wikipedia, "Spotify — Wikipedia, L'enciclopedia libera." [Online]. Available: <https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge/#rules>
- [4] S. Giallorenzo, "Measures and Metrics, Nodes." [Online]. Available: <https://saveriogiallorenzo.com/teaching/na/slides/L05.pdf>
- [5] T. South, M. Roughan, and L. Mitchell, "Popularity and centrality in Spotify networks: critical transitions in eigenvector centrality," *Journal of Complex Networks*, vol. 8, no. 6, Dec. 2020, doi: [10.1093/comnet/cnaa050](https://doi.org/10.1093/comnet/cnaa050).
- [6] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," *Advances in Physics*, vol. 56, no. 1, pp. 167–242, Jan. 2007, doi: [10.1080/00018730601170527](https://doi.org/10.1080/00018730601170527).