Community Detection in Spotify Playlists

Brandon Tran, Dare Hunt, Andrew Moktarzadeh, Junjie Li

March 2023

1 Abstract

Recent work introduced the vast unfolding of community detection in large networks, in which a heuristic methodology not only identifies communities, but also measures the density between nodes in modules that highlight the strength of a subcommunity. It was shown that such clustering methodologies can facilitate community detection with the benefits that include exceeding other clustering algorithms in time complexity. In this paper, we focus on using CESNA (Communities from Edge Structure and Node Attributes) in order to combine both methodologies to see if we can categorize music artists by genres based on the Spotify playlists they are in. By using a combination of the generated network's edges and the attributes of each artist, CESNA was able to construct 7 distinct communities with 53% of artists in each community matching the top genres (such as indie-pop) in each respective community and 82% of artists matching the general genre of each community.

2 Introduction

Networks are everywhere, from the social interactions of our daily lives to the complex systems of technology and infrastructure that underpin modern society. However, while basic concepts of network analysis can be traced back to the 18th century (when Leonhard Euler published one of the first known papers on graph theory) [2], it was not until the last few decades that advances in computing power and data collection techniques have made it possible to analyze large-scale network data in a systematic and rigorous manner. Today, network analysis is a vital field of research with applications in fields ranging from biology to social science to computer science. By structuring data in such a way that nodes are linked together by edges if they share a relation (much like how users on a social network are connected if they are friends), new analysis techniques can be used to take advantage of this unique data structure.

When analyzing the information within various data networks, communities form naturally within them through connections between individual points of data, or nodes. However, as more individual nodes of data are added to the

data collection, the number of connections between nodes and the number of communities formed to represent these connections grows exponentially, creating difficult problems to overcome when analyzing the data in a timely manner.

3 Related Work

It's important to note that grouping data has been a challenge that we have been trying to solve, and can be achieved using "clustering" algorithms, where using multiple attributes for each data entry can be used to find similarities and differences between them to create "clusters". However, the idea of locating and recovering communities is focused on networks as analysis largely relies on a key component of the data structure - the edge. This is where the idea of the planted clique problem was first presented: identifying the subset of nodes in a network that have something in common, determined by the density of edges in a network. The challenge was constructing an algorithm to do so that could perform in efficient time. Methods to achieve this in polynomial time were introduced in 1995 by Luděk Kučera [4], and improved upon in 1998 by Alon, Krivelevich and Sudakov [1]. Both proposed constraints to the size of the planted clique relative to the network, where the planted clique could be found with high probability. More recently, the paper "Computational Lower Bounds for Community Detection on Random Graphs" [3] observes that there are calculations to clearly define three bounds that determine the level of difficulty to retrieve a planted clique: simple, hard, and impossible. These bounds are based off the size of the network and size of the planted clique (or community) is. This prior research exposes a drawback in graph data, given that some situations cannot be optimized at all.

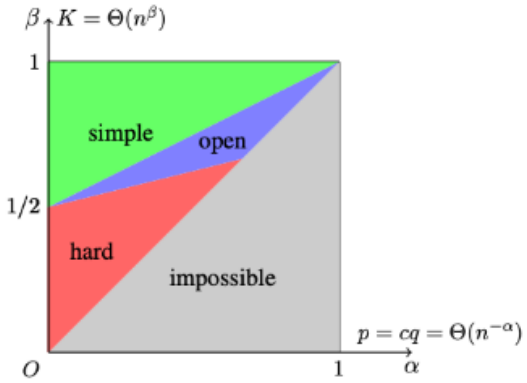

Figure 1: The general bounds describing the difficulty of a given community detection problem [3].

Traditional approaches for community detection, such as the planted clique problem, have focused on identifying dense subgraphs in the network. However, these methods may not be optimal for networks with overlapping communities or complex features. This has led to the development of alternative approaches, such as CESNA, that leverage additional information beyond the network topology. By considering both edges and features, CESNA can provide a more nuanced understanding of the communities in the network, which is essential for many real-world applications.

CESNA takes a different approach for detecting communities by using a combination of the network architecture as well as the specific features of each node. By considering both sources of information from the data independently, CESNA is able to better understand and assess which nodes best belong in each community. While there do exist other algorithms which take both edges and features into account, CESNA is able to outperform them when dealing with networks that feature overlapping communities, an important distinction given how interconnected some communities can be in networks.

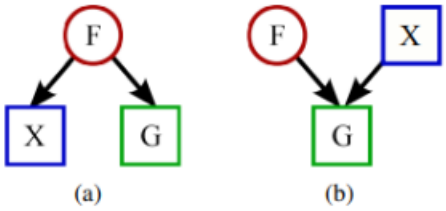

Figure 2: Two ways of modeling the statistical relationship between a graph G, attributes X, and Communities F. Circles represent variables that need to be inferred while squares represent observed variables.

There are four method classes for community detection in networks that we will use in comparison to CESNA. First is the Heuristics method class which refers to the set of algorithms that identify communities in a network using simple rules. The most popular heuristic method is the Girvan-Newman algorithm, which repeatedly removes edges with the highest connectedness centrality (number of shortest paths in the network that pass through an edge) until the network is divided into communities. Second is the LDA (Latent Dirichelet Allocation) which is used to model the distribution of nodes based on their connections to other nodes where the nodes are treated as "documents" and the edges are treated as "words". Third is the Clique-based heuristics class which is a set of algorithms that identify cliques or subsets of nodes that are fully connected to each other in a network. Nodes belonging in the same clique are most likely to belong to the same community. Lastly, the social circles method which is based on the social network concept that people who share common interests form social circles where nodes are connected based on the similar attributes or

connections.

Method Class	O	H	D	N
Heuristics	✗	✓	✗	100,000
LDA-based	✓	✗	✓	85,000
Clique-based-heuristics	✓	✓	✗	100,000
Social circles	✓	✓	✗	5,000
CESNA	✓	✓	✓	1,000,000

Table 1: Comparison of method classes and CESNA. O: Detects overlapping communities, H: Assigns hard node community memberships, D: Allows node attributes, N: Largest network processed in 10 hours.

From this table, we see that CESNA can detect overlapping communities, assigns hard node-community memberships, allows dependence between the network and node attributes, and the most efficient method class for community detection in networks with node attributes.

4 Data Description

Our primary source of data is a dataset of Spotify playlists collected by Andrew Maranhão, which is freely available on Kaggle [6]. This dataset was collected using a subset of users who published their nowplaying tweets via Spotify. This tabular dataset lists a row for each song that was tweeted out, containing the name of the song, the artist of the song, and the playlist that song was playing from. While the data also contained the user IDs of each person who tweeted the track they were listening to, our model does not take personal information as input.

Dataset	# of Artists (nodes)	# of Playlists
Raw	290,002	161,530
W/out Missing Vaues	289,784	161,345
Artist in more than 1 Playlist	148,690	158,880
Random Sample with above conditions	20,396	2,000

Table 2: Breakdown of Dataset

We began our initial analysis of the dataset by removing any rows that contained missing information as well as playlists that only contained a single artist.

We then randomly sampled 2000 playlists from our dataset, and increased the threshold to only keeping artists that were in at least 10 playlists. A graph was then generated by iterating through artists within playlists and generating edge weights between artist nodes. Initially, this gives us a novel perspective on how artists are connected through their appearances in playlists.

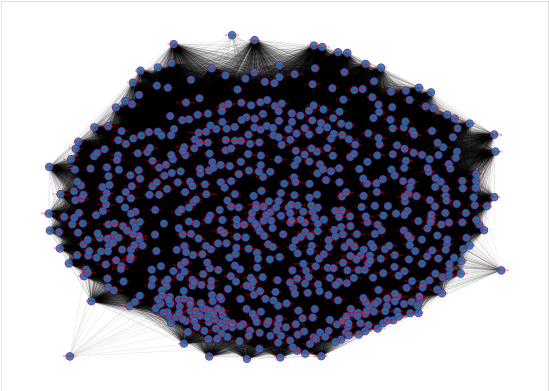

Figure 3: Visual insight into how artists are connected through their appearances in playlists.

5 Methods

On a technical level, in network G, each node has K attributes, and there are C communities in total. The node attributes are represented in a binary matrix as X where $X_{u,k}$ is the k-th attribute of node u. Additionally, we consider a community membership matrix F, where we assume that each node u has a non-negative affiliation weight $F_{u,c} \in [0, \infty)$ to community c. If $F_{u,c} = 0$ then node u does not belong to community c.

For implementation, the adjacency matrix is constructed from nodes defined to be artists, where an edge is denoted by two artists existing in the same playlist. Our node attribute is constructed by calculating total playlist appearances by artist and identifying the top 25% threshold of appearance count. The attribute is the binary representation of whether that artist's total appearances is above the threshold.

The CESNA algorithm infers communities through four assumptions:

- Nodes that belong to the same communities are likely to be connected to each other
- Nodes can belong to multiple communities
- Nodes with more communities in common are more likely to be connected than those with less in common

- Nodes in the same community share common attributes

These assumptions are satisfied because: 1. edges are created from shared community memberships (edges are created if artists are in the same playlist), 2. Each node represented in the community membership matrix F is considered an independent variable which allows a node to belong to multiple communities, 3. Each member in community c has independent connections, so those with more communities in common are more likely to be connected than those with less, and 4. We can predict the value of each node's attributes based on a node's community membership. The objective function we are trying to solve is then $\hat{F}, \hat{W} = \text{argmax } \mathcal{L}(G) + \mathcal{L}(X)$ where $\mathcal{L}(G) = logP(G|F)$ and $\mathcal{L}(X) = logP(X|F, W)$. We break $\mathcal{L}(G)$ and $\mathcal{L}(X)$ into two subproblems by fixing community memberships F and weights W for each node and update F and W at the end of each iteration of gradient ascent.

6 Results

Using the CESNA algorithm, we identified seven distinct communities from the Spotify dataset. To assess the accuracy of our algorithm, we evaluated the top three genres in each community and computed the percentage of nodes within each community that have any of these three genres. We also used generalized genre for our accuracy metric here because there are several sub-genres of a main genre for example indie-pop and dance-pop which we believe that including all under the term pop is appropriate.

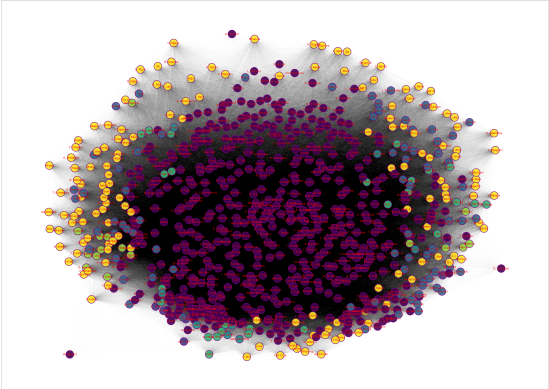

Figure 4: The same network, now assigning a color to each artist based on which community CESNA found them most likely to be in.

The first community detected belongs to the rock genre, containing 496 edges or playlists and includes popular bands such as Bon Iver and One Direction. The

genre accuracy of this community is approximately 45.4%, while the accuracy of the generalized genre improved to 78.8%.

The second community we identified is a dance pop genre group with only five edges, including artists such as Charli XCX, Demi Lovato, will.i.am, Naughty Boy, and The Pussycat Dolls. Given the small size of this community, both the genre accuracy and the generalized genre accuracy are 100%.

The third community is also a dance pop genre group with 76 edges, including artists such as John Mayer and Adele. The genre accuracy of this community is 60.5%, and the accuracy of the generalized genre is about 87%.

The fourth community is a hip hop genre group with only eight edges, featuring popular artists like Kendrick Lamar, Drake, and Lil Wayne. The genres of this community include hip-hop, rap, and pop. The genre accuracy of this community is 75%, and the accuracy of the generalized genre improves to 87.5%.

The fifth community is another dance pop genre group with 26 edges, including artists such as Britney Spears and Snoop Dogg. The genres of this community include dance pop, pop, and pop rap. The genre accuracy is 69.2%, and the generalized genre accuracy is 88.5%.

The sixth community is the second community in the rock genre group with a generalized genre of rock which has 17 edges including artists such as Beck and Red Hot Chili Peppers. The genres of this community include rock, modern rock, and pop rock. The genre accuracy of this community is about 65%, and it improves to 94% when we generalize the genre.

Finally, the last community we detected in our model also belongs to the rock genre with a generalized genre of rock. It has a community size of 148 edges including artists such as Coldplay and Arctic Monkeys. The genres of this community include rock, pop, and dance. The genre accuracy of this community is about 68%, and the generalized genre accuracy is 86%.

Overall, the total accuracy of the original genres associated with each artist is 53%, which improves to 82% when we generalize the genres. Looking at a single data point in its original form, we are only able to identify the playlists an artist is featured on. Using this network we can quickly identify several similar artists, not just limited to those in the same playlists. In a practical sense, a network like this can be used to generate new playlists or discover similar artists.

7 Conclusion

In conclusion, our study explored the use of CESNA algorithm for community detection in a network of music playlists. By leveraging both edge structures and node attributes, we identified seven distinct communities where the nodes shared common genres. Our evaluation of the accuracy of the communities showed that most communities achieved 60% or higher accuracy, with an overall improvement from 53% to 82% when generalizing the genres. These results suggest that CESNA can be a useful tool for identifying communities in dense networks with basic attributes.

Moreover, our study highlights the importance of considering node attributes

in community detection algorithms. By using node attributes such whether an artist was within top 25% of artist appearances, we were able to identify communities of playlists that are not only structurally similar but also share similar musical characteristics. This approach could be extended to other domains where nodes have additional attributes, such as user preferences or geographical location, to better capture the underlying communities in a network.

Finally, our preliminary work shows that given a dense network and a few basic attributes, CESNA was able to achieve a good accuracy score. This promising result suggests that we are reaching a new turning point in community detection in growing, denser networks. As networks become increasingly complex and multi-dimensional, community detection algorithms that can effectively leverage both network structure and node attributes will become increasingly important. We hope that our study will inspire further research in this direction and contribute to the development of more powerful community detection methods.

References

[1] Noga Alon, Michael Krivelevich, and Benny Sudakov. "Finding a large hidden clique in a random graph". In: *Random Structures & Algorithms* 13.3-4 (1998), pp. 457–466. DOI: https://doi.org/10.1002/(SICI)1098-2418(199810/12)13:3/4<457::AID-RSA14>3.0.CO;2-W. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291098-2418%28199810/12%2913%3A3/4%3C457%3A%3AAID-RSA14%3E3.0.CO%3B2-W.

[2] Stephan C. Carlson. *graph theory*. 2022. URL: https://arxiv.org/pdf/1406.6625.pdf.

[3] Bruce Hajek, Yihong Wu, and Jiaming Xu. *Computational Lower Bounds for Community Detection on Random Graphs*. URL: https://arxiv.org/pdf/1406.6625.pdf.

[4] Luděk Kučera. "Expected complexity of graph partitioning problems". In: *Discrete Applied Mathematics* 57.2–3 (1995), pp. 192–212. ISSN: 0166-218X. URL: https://doi.org/10.1016/0166-218X(94)00103-K.

[5] Martin Pichl, Eva Zangerle, and Günther Specht. "Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name?" In: (2015).

[6] *Spotify Dataset*. 2015. URL: https://www.kaggle.com/datasets/andrewmvd/spotify-playlists.

[7] Jaewon Yang and Jure Leskovec. "Overlapping Community Detection at Scale: A Nonnegative Matrix Factorization Approach". In: (2013). URL: https://cs.stanford.edu/people/jure/pubs/bigclam-wsdm13.pdf.

[8] Jaewon Yang, Julian McAuley, and Jure Leskovec. "Community Detection in Networks with Node Attributes". In: (2013). URL: https://www.kaggle.com/datasets/andrewmvd/spotify-playlists.