Application of Graph for Music Discovery

Delisha Azza Naadira 13519133
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13519133@std.stei.itb.ac.id

***Abstract*—Discovering new music and artists is a good way to keep your listening habit fresh. With the rising popularity of streaming services and digital music distribution, it is much easier for music listeners to broaden their horizon in music. There are several ways to discover new music, and a few of them uses graphs to determine which music to recommend based on the music you like. One way of discovering new music is by using a self-adapting system that determines what music someone might like based on the artist name they enter.**

***Keywords*—Artist, Discovery, Graph, Music**

I. Introduction

Discovering new music and artists is something that some people like to do, especially when they are getting bored of listening to the same things for a certain amount of time. For people who like to cross musical genres and give different music styles a try, they will get their recommendations by listening to playlists of those said genres or by looking at music forums and reviews. But, for people who like to listen to new music that are similar to what they already listen to, they can get their recommendations with algorithms and artist similarities.

In this day and age, people can easily listen to music through streaming platforms like Spotify, YouTube, Apple Music, etc. These streaming platforms usually provide users with an extensive music catalog, spanning across genres and decades. Not only that, they will also give recommendations about other music the user might also like. The recommendations they give are based on the user's streaming history, and they will compare it with other users' listening history. This method of recommendation uses Collaborative Filtering (CF), which, rather than filtering and recommending something based on similarities between items, it is based on similarities between users. In this case, they will compare between users' listening habits.

Another way to discover new music that is less personalized but still unique is by using music search engine where people can put an artist's name and it will recommend them other artists they might like. Global Network of Discovery (GNOD) is described as "a self-adapting system that learns about the outer world by asking its visitors what they like and what they don't like". There are two music discovery projects in GNOD, Gnoosic and Music-Map, and they give people different music discovery experience. It's an ever-growing site that will add more artists and update the probabilities as time goes along.

II. Graph

A. Definition

A graph (denoted as $G = (V, E)$) is a structure consisting of a non-empty set of vertices or nodes (V) and a set of edges (E) that connect one node to another. Two nodes are adjacent if they are connected by an edge, and a node is incident to an edge if the edge connects the node to another node. A degree of a node is the number of nodes adjacent to it, or the number of edges that are incident to it.

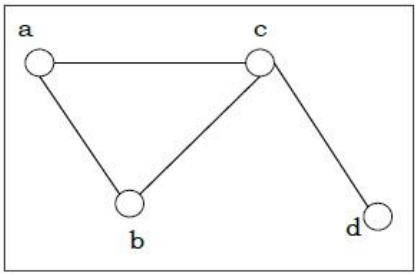

Figure 1. Graph. Source: *https://www.tutorialspoint.com/*

B. Types of Graphs

1. Simple Graph
   A simple graph is a graph that does not have any loops and multiple edges.

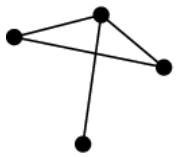

Figure 2. Simple Graph. Source: *https://mathworld.wolfram.com/*

2. Multi-Graph
   A multi-graph is a graph that has multiple edges connecting two nodes.

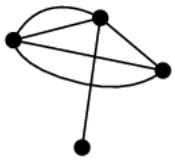

Figure 3. Multi-Graph. Source: *https://mathworld.wolfram.com/*

3. Pseudo-Graph
   A pseudo-graph is a graph with loops. A loop in a graph is an edge that connects the same node.

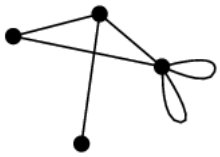

Figure 4. Pseudo-Graph. Source: *https://mathworld.wolfram.com/*

4. Directed Graph
   A directed graph is a graph where the edges have directions. The edges are represented by arrows, meaning the nodes are related in one direction.

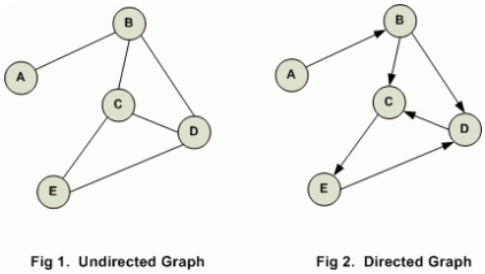

Figure 5. Undirected and Directed Graph. Source: *https://algorithmsinsight.wordpress.com/*

5. Null Graph
   A null graph is a graph with no edges connecting the nodes.

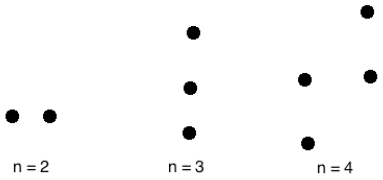

Figure 6. Null Graphs. Source: *http://mathonline.wikidot.com/*

6. Connected Graph
   A connected graph is a graph where there is at least one path to get from one node to another.

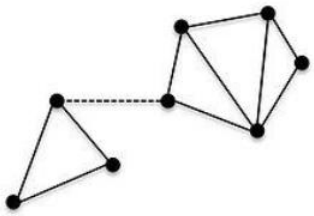

Figure 7. Connected Graph.

7. Weighted Graph
   A weighted graph is a graph where each edge is given a numerical weight.

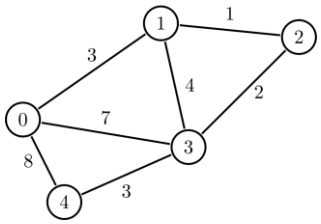

Figure 8. Weighted Graph. Source: *https://hyperskill.org/*

8. Complete Graph
   A complete graph is a graph where each node is adjacent to every other node. The degree of each node of a complete graph with *n* nodes is *n-1*.

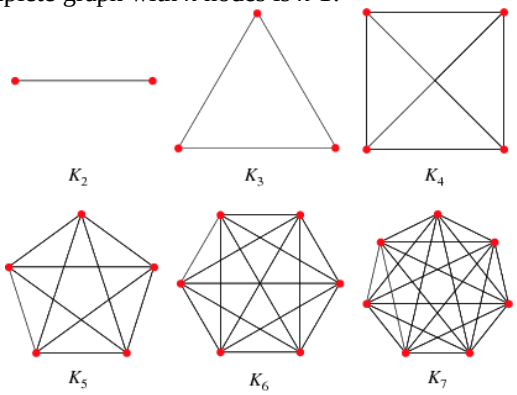

Figure 9. Connected Graphs. Source: *https://mathworld.wolfram.com/*

9. Cycle Graph
   A cycle graph is a graph where each node is connected to two other nodes. Each node in a cycle graph has a degree of two.

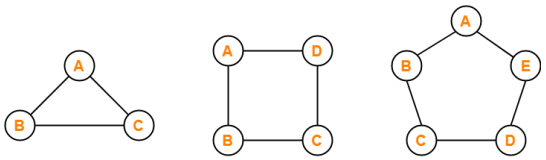

Figure 10. Cycle Graphs. Source: *https://www.gatevidyalay.com/*

10. Regular Graph
    A regular graph is a graph where every node is connected to the same number of nodes. The degree of every node in a regular graph is the same. A regular graph where every node has a degree of *n* is called an *n*-regular graph.

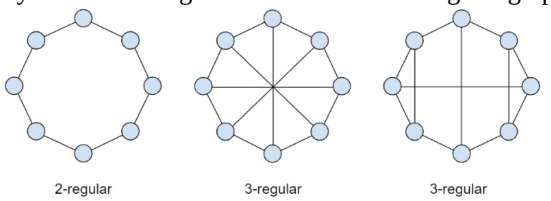

Figure 11. Regular Graphs. Source: *https://thomasvilhena.com/*

11. Bipartite Graph
    A bipartite graph is a graph where the set of nodes can be split into two disjoint sets *V1* and *V2*, and that each edge connects a node from *V1* to a node from *V2*.

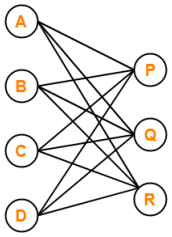

Figure 12. Bipartite Graph. Source: *https://www.gatevidyalay.com/*

C. Representation of Graphs

1. Adjacency Matrix

The adjacency matrix of graph G with n nodes is the $n \times n$ matrix $A = [a_{ij}]$, where a_{ij} represents the adjacency between nodes i and j. If nodes i and j are adjacent, $a_{ij} = 1$, and if not, $a_{ij} = 0$.

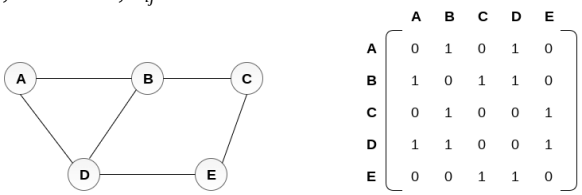

Figure 13. Undirected Graph Adjacency Matrix. Source: *https://www.javatpoint.com/*

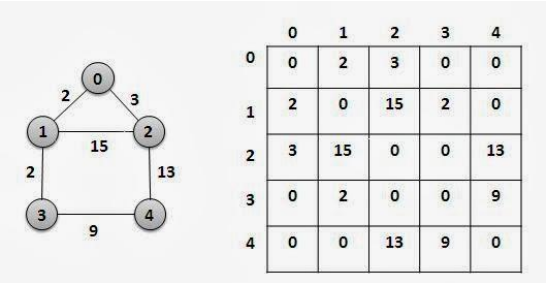

Figure 14. Weighted Graph Adjacency Matrix. Source: *https://www.thecrazyprogrammer.com/*

2. Incidence Matrix

The incidence matrix of a graph G with n nodes and e edges is the $n \times e$ matrix $A = [a_{ij}]$, where a_{ij} represents the incidence between node i and edge j. If node i is incident to edge j, $a_{ij} = 1$, and if not, $a_{ij} = 0$.

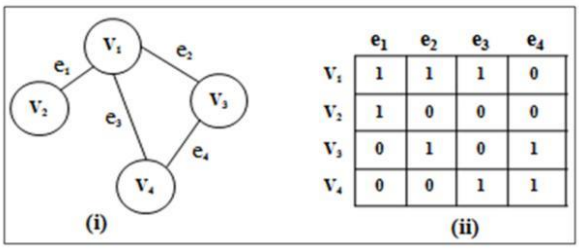

Figure 15. Undirected Graph Incidence Matrix. Source: *https://www.researchgate.net/*

3. Adjacency List

The adjacency list of a graph G is a list of nodes in the graph and their adjacent nodes. Adjacency list can be represented with a table or an array.

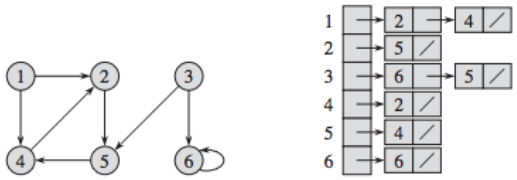

Figure 16. Directed Graph Adjacency List. Source: *https://notes.shichao.io/*

III. GNOD'S MUSIC DISCOVERY

A. Projects

Global Network of Discovery (GNOD) is a discovery system that gives people recommendations based on the user's input and information from past users on music, art, literature, movies, and products. As the creator describes it: "Gnod is a self-adapting system that learns about the outer world by asking its visitors what they like and what they don't like." For music discovery, they offer two ways of recommending music, the "Discover" project and the "Music-Map" project.

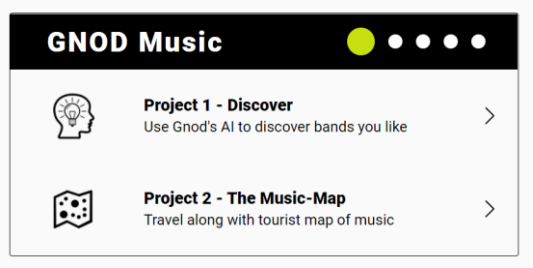

Figure 17. GNOD's Music Projects. Source: *https://www.gnod.com/*

B. Gnoosic Discover

The first music discovery feature in GNOD's website is the Discover project, also called Gnoosic. When clicking on it, the site will ask the user three bands they already know and like. GNOD's creator wrote: "Gnod is kind of a search engine for music you don't know about. It will ask you what music you like and then think about what you might like too."

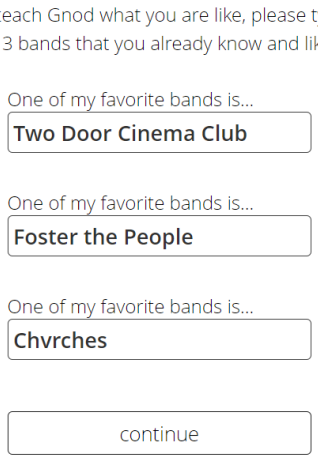

Figure 18. Gnoosic asking for the names of three artists. Source: *https://www.gnoosic.com/*

After entering three names of artists, the site will predict other artists that the user might like based on the three artists they entered. User can also say whether they like the artist the website recommended or not, or if they don't know the artist. The site sometimes also gives a sample of the artist's music. The user can also easily look for the artist in streaming platforms they have and have a listen to them to find out whether they like the artist or not. Usually, Gnoosic will give the user artists that are more popular, and then get more obscure as it goes on, from artists people are familiar with to artists that are less popular.

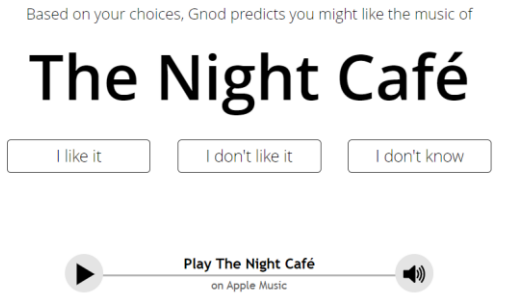

Figure 19. Gnoosic predicting another artist the user might like.
Source: *https://www.gnoosic.com/*

After going through a cycle of artists, Gnoosic will give a summary of artists that were recommended and the user's opinions on them.

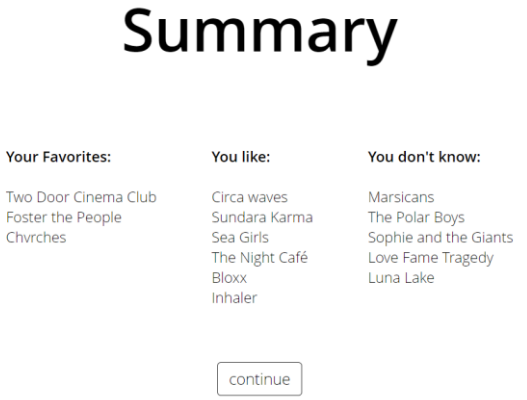

Figure 20. Gnoosic giving a summary of artists.
Source: *https://www.gnoosic.com/*

C. Music-Map

The other project for GNOD's music discovery is the Music-Map. When clicking on the Music-Map project, the user gets to enter the name of an artist they like. After entering it, the user will be given a bunch of other artists related to the artist the user entered in a map, where the closer the names of the artists are, the site thinks the two artists are related more, and that probability that the user might like the same artists are higher. They can also click on other artists on the map and the site will give the map of that artist. As the user go along through the artist maps, they will discover a lot of other artists they can listen to.

Figure 21. Music-Map when user enters "Foster the People".
Source: *https://www. music-map.com/*

IV. Application of Graph in Music Discovery

Both projects in GNOD's music discovery implements the use of graph—more specifically, weighted graph. The Music-Map project is a lot clearer in displaying the graph with the use of a map, and it's only used to display the graph. Meanwhile, Gnoosic uses a weighted graph to predict recommended artists, but it also collects data from the users' inputs and add more value to the weights of the edges.

A. Graph Representation

In a weighted graph, nodes are connected to edges where each edge is given a value. In the GNOD music discovery projects, nodes represent names of artists and edges represent the number of people who like the same two artists. The weighted graph in the figure below means that 58 people like both artist 1 and artist 2, 24 people like both artist 1 and artist 3, 12 people like both artist 2 and artist 4, etc.

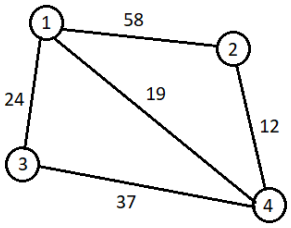

Figure 22. Example of a weighted graph.

In the actual application, there are now hundreds of thousands of nodes and millions of weighted edges, depending on how big their database is. Each node can also be connected or not, depending on what the users have inputted in the past.

B. Application of Graph in Gnoosic Project

The data of artists in both the Music-Map and Gnoosic projects were collected from people who were using Gnoosic. According to the creator of GNOD, when the site first started off the database was completely empty, but as time goes along and more people have used the site, the database is more complete.

Firstly, the user will enter three artist names. Then, the value of the weights between the three artists' edges in the database will be added by one. If the user enters an artist's name that isn't in the database, the system will add another node in the graph and connect them to the other two artists' nodes with the weight value of 1. The system will then look for other nodes that have the biggest weight value with those three artists. When an artist's name is given and the user says they like the artist, the value of weight between the artist and the three other artists will be incremented by one. The system then finds another node, and it happens until the cycle ends.

Below is an example of a weighted graph with 7 nodes after a user enters the names of artists b, d, and f and the values between the edges connecting those three artists are incremented by one. System then checks which node has the biggest sum of weights with the other three artists.

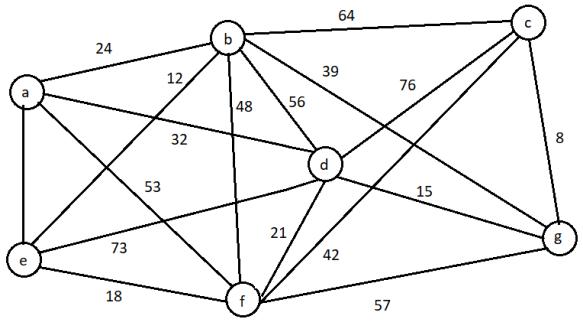

Figure 23. Example of a weighted graph with 7 nodes.

Node	Weight			Sum
	b	d	f	
a	24	32	53	109
e	12	73	18	103
c	64	76	42	182
g	39	15	57	111

Figure 24. Table of sum of weights between nodes of favorite artists and other nodes.

According to the table, artist c has the highest sum of weights, followed by artists g, a, and e. This means when the user enters the name of artists b, d, and f, artist c will be recommended first, then artist g, artist a, and artist e last. In a case that the user says that they like artist c, the weights of edges between artist c and artists b, d, and f will become 65, 77, and 43 consecutively.

Because of this, it can also be explained why the recommended artists become less and less in popularity. The reason is that less popular artists are less likely to be inputted on the search bar. Because of it, the weight of edges adjacent to the less popular artists are fewer than the weight of edges adjacent to the more popular artists.

C. Application of Graph in Music-Map

In the Music-Map project, weighted graph is also used, but Music-Map doesn't change anything with the database, just displays the graph in the form of an artist map. The user will enter an artist's name. Then, the system will look for the edge incident to the artist's node with the biggest weight. The node with the biggest weight will be the closest from the inputted artist when displayed. Then it will look for the next biggest weight of edge, and so on until it finds enough artist nodes to display. The scale of closeness depends on the database and the weights of edges incident to the node.

For example, take the weighted graph in *Fig. 23*. If a user enters the name of artist d, the system will look for the values of weights of edges adjacent to artist d.

Nodes	a	b	c	e	f	g
Weight	32	24	76	73	21	15

Figure 25. Table of weights of edges between artist d and other nodes.

Then, after comparing each weight, the system will display the map for artist d, which is visualized in the figure below.

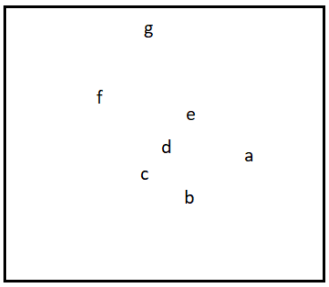

Figure 26. Map visualization from graph in *Fig. 23.*

D. Faults in System

The system in GNOD uses what users entered and adds them to the database so that the data can then be compared with. If an artist's name isn't in the database, users can suggest and vote on artist names. If it has enough votes, the system will add it to the database to be compared with other artists.

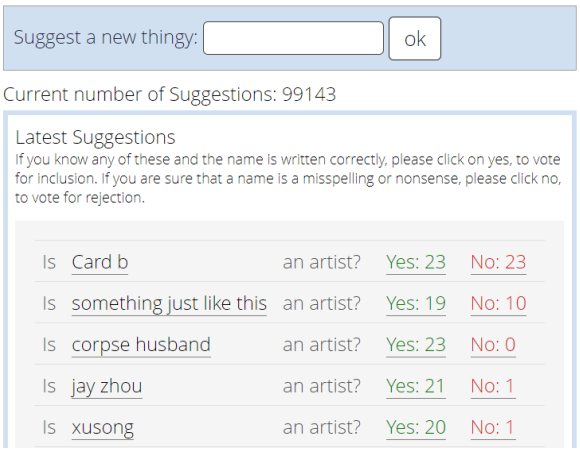

Figure 27. Gnoosic's suggestion page. Source: *https://www.gnoosic.com/*

Because of this, there are instances where an artist's name comes up more than once and the name isn't an artist's name (a song name or an album name, typographical error). This causes the data to not be as accurate.

The system is also not always accurate. It's not always guaranteed that Gnoosic's recommended artists will be to the user's liking. In the end it's all according to someone's tastes to like or dislike something. Users can help making the system more accurate by using it more and adding to the database more. Like the creator said, the site started off empty but as time goes along more and more people contribute to the site's database, and now there are tens of thousands of artists registered in the database.

V. CONCLUSION

A graph can be applied for many things. Graphs are not only used in natural sciences, but it's also used in computer science. In this case graphs are used to determine the relations between two artists in order to predict similarities between artist for discovering new music.

There are many ways to discover new music, one of them is by using a discovery system that's constantly self-adapting

based on what the visitors of the site input. The bigger value of the weight of an edge is, the higher the probability that someone likes both of the artists.

Despite this, the system still has its faults. But, with more contribution from people, the system and database will expand so that the results will become more accurate.

VI. Acknowledgment

The author would like to thank the professors of Discrete Mathematics, especially Mr. Rinaldi Munir for teaching about graphs and providing the knowledge that will be applied in the everyday life. The author would also like to apologize if there are any errors or misinformation in this paper.

References

[1] Munir, Rinaldi. *"Graf (Bagian 1): Bahan Kuliah IF2120 Matematika Diskrit"*. http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf, accessed in 10/12/2020.

[2] Munir, Rinaldi. *"Graf (Bagian 2): Bahan Kuliah IF2120 Matematika Diskrit"*. http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian2.pdf, accessed in 10/12/2020.

[3] Munir, Rinaldi. *"Graf (Bagian 3): Bahan Kuliah IF2120 Matematika Diskrit"*. http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian3.pdf, accessed in 10/12/2020.

[4] Gibney, Marek. *Global Network of Discovery (GNOD)*. https://www.gnod.com/.

[5] "*Graph and its representations*". https://www.geeksforgeeks.org/graph-and-its-representations/, accessed in 09/12/2020

[6] Hauskrecht, Milos. "*CS 441 Discrete Mathematics for CS: Graphs*". https://people.cs.pitt.edu/~milos/courses/cs441/lectures/Class25.pdf, accessed in 11/12/2020.

[7] "*Representation of Graphs: Adjacency Matrix and Adjacency List*". https://www.thecrazyprogrammer.com/2014/03/representation-of-graphs-adjacency-matrix-and-adjacency-list.html, accessed in 10/12/2020.

[8] Jacca-RouteNote. "*This map takes you to artists you will love based on music you already do*". https://routenote.com/blog/plot-journey-artists-youll-love-music-map/, accessed in 10/12/2020.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2020

Delisha Azza Naadira 13519133