Smartphone application for music recommendation based on musician network

Junqing Qian

Kongens Lyngby 2013
IMM-B.Sc-2013-5

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-B.Sc-2013-5

Abstract

This thesis is focused on presenting a system for recommending music to the smartphone users. The system takes the users favourite musician as input and generate recommendations based on this musicians's social relations. The approach is based on the social networks of the musicians. Each of the different networks for the musicians are constructed and analysed to provide possible recommendation results. The most significant findings are also presented in the application as overview and suggestions for interesting musicians.

This approach distingushes itself from the existing solutions and is implemented into a prototype to be tested for the functionality and potential market value. Users will be able to navigate through the musicians of interest and view their general information as well as recommendations based on these musicians. Each of the recommended musician includes an audio preview as presentation. The network approach can exceed the potential recommendation content of the content-based filtering and has potential to reach the widely used collaborative filtering in terms of recommendation diversity.

The prototype utilizes the web-based application framework to achieve cross-fit comptatibility on both Android and iOS platforms. The advantages and drawbacks of this framework is tested and discussed.

Finally a simplified version of user experience process and the gathered feedback validates the functionality and the potential market value of this application.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring an B.Sc. in Software Technology.

The thesis deals with the development of a recommendation system based on the networks of musicians. This recommendation system is then implemented to a smartphone platform.

The work on this thesis was done from February 6th 2013 to May 2013. The workload corresponds to 15 ECTS points. The supervisor of this thesis is Michael Kai Petersen.

Lyngby, 17-May-2013

Junqing Qian

Contents

Chapter 1

Introduction

This chapter provides insight to the background and motivation behind the thesis, as well as the visions and goals accompanying the project, which are based on the initial expectations prior to the project's completion. The desired outcome is also defined

1.1 Motivation

During the past decade mobile devices have become an integral part of the modern society. It has become a society standard to own and use a mobile phone on an everyday basis. The introduction of smartphones took mobile devices to another level. The accessibility of Internet as well as having practical hardware such as sensors built-in makes it a powerful computer that fits into the palms of the user. A device of this computing power and mobility offers endless possibilities for a developer to utilize. This creates a world of possibility for the independent developers to create simple yet innovative applications with great potential of commerciality.

Simultaneously with the development of mobile technology, information technology has also undergone advancement. The Internet has stored a vast amount of accessible information that would otherwise be tedious and inefficient to collect

manually. It is no longer necessary to visit the library and open books to search for information, the fact that "I will google that" is becoming a socially acceptable term to use indicates the dominance of the Internet within information technology.

Aside from technology, music plays a considerable role in the modern society. It is possible to assume that almost every individual who has access to the modern technology such as a smartphone, have a certain personal preference and interest in music. Each person gets in touch with the different style of music by self-exploration or recommendation by friends or other sources. It is also safe to assume that most individuals have not listened to every song that has ever been recorded or performed, in which case there exists a possibility to further explore for new music. However it is exhausting and frustrating to explore a complete unfamiliar area of music, the chances of the music lives up to the expectations of the user is at random. This thesis describes a logical sense of music recommendation based on users' preferable artists and navigates through their social networks, which has a higher possibility to contain elements similar or inspired from the users comfort zone.

Using the methods and tools from computer science, it is then possible to extract artists' information and analyse the chosen data. The results of the analysis can be visualized as musicians' connection to each other, whether they have collaborated together or have musical similarities with each other and finally the influences of a certain musician to others. Interpreting the connections between musicians can lead to many researches in the subject of musicology, sociology or possibly even understanding the music industry.

The focus of this thesis is to present the network connections and recommend songs based on a certain artist in the network. The results from the recommendation system will then be implemented into a prototype application for smartphones.

1.2 Project Goals

The first goal of this project is to construct a relevant dataset. The accuracy of the dataset directly impacts the analysis results and thereby the precision of the recommendation system.

The main goal of this project is to obtain and interpret results from the quantitative analysis of the musicians' networks, using network theory. The dataset should be processed and visualized. Each of the networks are inspected and

utilized to provide potential results for the recommendation system.

The final goal is to build a prototype smartphone application to present the recommendation and certain conclusions that might be interesting for the potential users. The prototype has to be functional on at least one smartphone platforms and will be tested by simple user experiences to validate the functionality and market value of this application.

1.3 Thesis Outline

This thesis is divided into several chapters; each chapter presents a different process of this research.

The first chapter describes the background and motivation of this thesis, as well as presenting the goals to complete through out the project.

The second chapter is focused on investigating the existing recommendation systems and present the necessary tools and theories within software development and network analysis

The third chapter is focused on constructing the dataset and the networks to enable further design of the recommendation approach. The design of the prototype is presented by defining the tools, platforms and simple mock-ups.

The fourth chapter describes the implementation process. The detailed data collection and processing are presented as well as the construction of the different networks. In-depth analysis is conducted on these networks and the most significant findings are defined. Finally the process of implementing the prototype is explained.

The prototype is tested in the fifth chapter. Several use cases are constructed to test the prototype's functionality. Results from a simplified user experience investigation are presented and discussed.

Finally the last two chapters focus on reflecting the accomplishments and discuss the possible issues and solutions of result accuracy. Subjects of further research are also presented.

Chapter 2

Analysis

This chapter presents the existing recommendation approaches and investigate the popular music related applications that utilizes these recommendation systems. Furthermore the aspects of network science and its tools are described.

2.1 Recommendation Systems

The subject of recommendation is a subclass of information filtering system. [7] The recommendation system helps the users to discover new content of interest and the musicians to gain audience. Successful recommendations benefit both partners in this relationship. This explains the gained popularity and even demand of developing appropriate approaches to recommendations. The existing approaches can be divided into two major subjects:

- Content-based filtering
- Collaborative filtering

The two approaches use different sources to gain information of the possible contents to recommend to the user.

2.1.1 Content-Based Filtering

The content-based filtering assigns attributes to the content by its own properties. Each song or artist is treated as comparable objects. The recommendation content is based on the similarity of the attributes between two objects. This item-based comparison constantly seeks similarity and therefor limits the recommendation output. Therefor decreasing the likelihood of discovering into new regions of music, thus it is often considered and criticized for lower recommendation value.

2.1.2 Collaborative Filtering

The collaborative filtering counters this problem by creating user-based models to seek out behaviours or patterns of the user or users with similar profile. This approach arguably offers more accuracy, because it is based on empirical data of the user and is capable to introduce the less familiar musical territory for the user. The disadvantage of the collaborative filtering is that it requires a reasonable amount of information before the system can provide reliable results, in which case it means it is problematic to start up this system. A phenomenon referred to as cold start.[17]

2.1.3 Usage

Different companies utilize and specialize in the different approaches, Pandora or EchoNest are they most known names for the content-based filtering. Pandora uses over 400 attributes provided by the Music Genome Project and attaches them to songs and artists while EchoNest conduct text- and acoustic analysis to compare songs to seek similarity.[10] On the other hand, Last.fm utilizes collaborative filtering to observe the users behaviours and conduct analysis on them to create recommendations. [13]

While the two approaches focus on different aspect to create recommendations, it is also possible to combine the two and creating a more accurate system to cover the drawbacks of each system. Netflix is an example to utilize both filtering methods and focus on creates accurate recommendations for movies.[6] It is also reasonable to assume that the recommendation systems in the most popular music apps such as Spotify or iTunes would contain a complex algorithmic structure, utilizing both approaches for optimal output.

2.1.4 Social Network Approach And Comparison

The two dominant approaches of recommendations both require a reliable and sufficient dataset to function. The content-based filtering requires professional analysis for each song and attaches precise attributes throughout the whole dataset. The collaborative filtering would require recording users' behaviour over a certain amount of time to create usable statistics. Both methods present challenges for independent developer.

The goal of this project is to utilize the characteristics of musicians' social networks to develop an approach to recommend music. As there is no user behaviour model and statistics, this approach does not associate with collaborative filtering. A uniform user model is applied to this approach – the users who are interested in a certain musician's work are also, to a certain degree, interested in the work of this musician's social acquaintances. If this requirement is fulfilled, then it is possible to create area of interest that differs from the two dominant approaches. The area of interested created with this approach will cover music territories within the user's comfort zone as well as stretching to the more unfamiliar regions. This feature distinguishes itself from the traditional content-based filtering, as the recommended content is not necessarily similar to the user's musical preferences. The content in this case is the musician and the attributes to attach to the musician are then his/her social networks. This specific feature contains advantages and disadvantages that require further considerations.

Below is a simple illustration that roughly estimates the areas of possible recommendations by the different approaches.

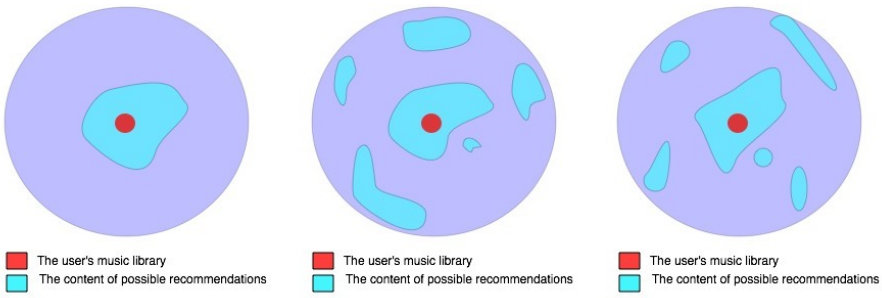


Figure 2.1: The first graph is the content-based filtering, the second graph is the collaborative filtering, the last graph is the network approach

The content-based filtering contains the most compact area as recommendations, as it constantly compares songs or artists with similarity. The estimated area

creates a large region revolve around the users preferences as centre, in which case results are very likely to be appreciated by the user but however is less capable of exploring into new areas of music.

The collaborative filtering counters this problem by utilizing other similar user's statistics to predict new areas in music that still would be likely for the user to acknowledge.

The network approach has the potential to exceed the content-based filtering and reach the collaborative filtering's diversity in recommendations. However the estimated areas are only based on musicians' relations, therefor the areas are simply just point of interests with random chance of falling into the musical taste of the user. Thus the definition of which area are potential recommendation materials is crucial. Each area of interest should be examined in the network data to reach certain degree of benefit to the recommendation system before conducting them into the results.

The most considerable drawback with the network approach is the compromised desirability of the results as they might cross into territories the user is not comfortable with. However, depending on the individual's definition of music discovery, unexpected results are not always equal to pointless results. Even if the result does not satisfy the user, it is still constructed based on a relevant point of interest.

2.2 Software Related Analysis

The recommendation approach is aimed to be implemented as a functional prototype for a smartphone. This section describes the thoughts and analysis before beginning to develop the prototype.

2.2.1 Usability

The usability of a smartphone application is crucial for its chance to succeed within the increasingly demanding user market. Developers often prioritize the functionality during the development, and overlook the trivial user experiences. The application should focus on the usability of the application to increase its value of desirability on the market.

The interface should provide clear overview of the information and functional-

ity. In which case it means all non-text contents should be distinguishable in such degree that the user does not need assist to understand its property. All interactive objects should either be self-explanatory or have informative instructions. Even though texts are useful to assist the user in the usability experience, it is most beneficial to keep the text compact. The wall of text is strictly opposed to user friendliness. Navigation wise, it is most favourable to show clear data depth and provide options to easily access the most useful data directory. This prevents the possibility of the user to get confused or disoriented during navigation.

2.2.2 User Feedback

Conducting user experience and get feedback on the design is an important approach while developing a software product. To fully understand the user experience it is preferable to conduct private interviews with potential users with different background and construct a set of tasks for the users to perform. The developer can then observe the users interaction with the application and gain insight of how the different users accomplish tasks in different ways. Then it is possible to acquire feedback for the general impression and improvement areas.

2.2.3 Market Analysis

There are various music related applications in the market. Most of the apps serve the main purpose of turning the smartphone into a music player device or radio. Here is a list of popular music apps:

- Spotify
- Pandora
- TuneIn Radio
- Last fm

All the listed applications are music players, which all include functionalities such as streaming music and create recommendations.

When expanding the searching criteria to include social networks, the results contains music oriented social networks such as Flow*d. These applications have

similar social networking structures as Facebook or Twitter while using music as the main theme.

This observation indicates that there is no apparent music related applications on the market that creates recommendation based on the musicians' social networks. Thus there is no apparent competition in the same category. However the empty spot in the market does not necessarily mean the opportunity of building a successful application. The purpose of the application require further research on the potential users before the conclusion can be drawn on whether this application have a certain market value.

It is useful to inspect the essential functionalities of the successful apps, as they contributed to their market value. Considering the recommendation function within these apps, the most noteworthy function is to be able to immediately listen to the recommendation content.

2.2.4 Platform

There are several existing smartphone platforms. The most popular platforms are Android, iOS and Windowsphone. Each of the platforms has a distinct environment system while maintaining certain similarities. The Android and iOS dominates the market by a combined market share of smartphones of 91% (Android 70%/iOS 21%)[5]. These two platforms also share similar application outline. Based on this, many commercially successful applications are designed to be compatible on both platforms to gain maximum user exposure.

2.3 Social Networks And Their Properties

Network science is a framework to study the interaction patterns between objects. It can be applied to the social interactions between individuals to investigate in different patterns of social behaviours. Large social networks are characterized by seemingly chaotic and irregular patterns of connections between the individuals. However they can be analysed by utilizing the tools of network theory and form understanding of its less obvious values. This thesis focuses on analysing the musicians' social networks to seek out patterns and potential recommendation contents to be presented in the smartphone application.

2.3.1 Basic Network Visualization

Social networks can be visualized by graphs containing a set of nodes to represent the individuals and edges to represent the relations between them. The size of the graph is defined by counting the number of nodes. Each edge can contain weight scores to further specify the relation of two nodes, creating a weighted graph. Not all connections are reciprocal, some relations are asymmetrical, to indicate the specific direction of a relation between two nodes. The first observation of a graph is to roughly estimate the connectivity. If a graph shows low or none connectivity, the nodes are either completely separated from each other or have very few edges to create small isolated groups. In this case it is very difficult to do an in depth analysis of the graph. On the other hand, when a graph shows one or several well-connected large components that contain a significant fraction of the total nodes, it is possible to apply mathematical models to further investigate for detailed results. However, while the expected network contains a reasonable size and complexity, it is usual for the initial graph to contain both elements – a central area with one or several isolated, tightly knitted components surrounded by a number of small, less connected groups. This scenario leads to the technique of extracting the giant component.

2.3.2 Extracting Giant Component

The giant component is the connected component which contains the largest fraction of nodes compared to other components. It is rare for two giant components to co-exist in a graph, which means the giant component is generally unique and distinguishable from all other components.[2] The giant component holds the qualities of being complex and dominant in graphs. Observing the other isolated components of the graph leads to the decision of whether it is preferable to exclude them while conducting measurements on the graph. Some measurements are only possible to calculate on a single connected component, such as the path length and centrality. If the graph contains several isolated components, then the approach is to conduct the measurements on each of the components separately. Comparing the results of the components leads to the general understanding of the graph. If the isolated components are reasonably small, the measurements of these components are negligible. Utilizing measurements from these small groups might distort the end results. Opposite to this, the qualities of the giant component make it most suitable to interpret overall key values of the graph. Therefor it is often preferable to extract the giant component before applying in depth analysis on the graph.

2.3.3 Mean Geodesic Length

Within a connected component, every node can reach every other node by following a path that goes through a set of nodes. The measurement of paths, also called geodesics, of the graph displays a certain level of connectivity and the mean distance between nodes. If the graph contains several components, the measurement of path length is conducted on each of the components, since a node from an isolated component cannot reach a node from another component, thus the length is infinite. The longest geodesic length is the diameter of the component. A prominent feature of a complex network is the so-called "small-world phenomenon", also known as "six degree of separation"[2], which suggest that the shortest paths are very small compared to the network size. It also implies that the mean geodesic length is approximately 6.

2.3.4 Degree Distribution

The degree of a node is the number of connections it has to other nodes, usually denoted as k. Then the degree distribution $P(k)$ describes the fraction of nodes in the network with degree k.

Many real world networks such as the worldwide web WWW and social networks tend to have a highly right-skewed distribution. This means the majority of the nodes have low degree while s small number of nodes, known as the "hubs", has high degree.

2.3.5 Assortativity Coefficient

This measurement is the Pearson correlation coefficient of degree between pairs of linked nodes.[14] Positive values of the measurement R indicate a correlation between nodes of similar degree. While negative values indicate relationships between nodes of different degree. The range of R value lies between -1 and 1, 1 being the perfect assortative network and -1 being the complete disassortative network.

2.3.6 Clustering Coefficient

A clustering coefficient is a measure of degree to which nodes in a graph tend to cluster together. It is a known property of most real-world networks, in

particular social networks, to contain tightly knit groups characterised by a relatively high density of ties.[2] This property is generally greater than the equivalent on a randomly established network.

The transitivity ratio is quantified based on the abundance of triangles in a network, also referred to as the global clustering coefficient.[15] Each triangle consist three closed triplets. The mathematical formula for transitivity ratio is defined as:

$$C = \frac{3 * NumberOfTriangles}{NumberOfConnectedTriplets} \tag{2.1}$$

It is also possible to determine local clustering coefficient for each individual node. This measurement is different for directed graph and undirected graph, based on the difference in the property of the edges between two nodes.

This value is calculated by the measuring the ratio of number of links within the direct connected neighbourhood and the number of possible links. The number of possible link for a node i is defined as *ki*(ki-1)* for the directed graphs and half of that value in the undirected graphs.

2.3.7 Centrality

Different nodes have different roles within the network based on their degree and position in the network. To determine the relative importance of different nodes in a graph, centrality measurements are utilized. There are four measures of centrality, each using a different approach and return different results. The choice of the measurement method should be based on the purpose of the application.

2.3.7.1 Degree Centrality

Degree centrality is the first and simplest centrality measurement. It is defined as the number of links incident upon a node, thus the degree of the particular node. A node with highest degree within the graph is the most central according to the degree centrality. The central nodes in this measurement have most immediate connections compared to other nodes, and can therefor directly affect a larger section of the network.

It is reasonable to assume that people with particularly many acquaintances can be looked as being important figures; however the degree centrality is primarily local in scope. Therefor the more advanced centrality methods can be applied to determine important nodes.

2.3.7.2 Closeness Centrality

The closeness centrality is based on computing the lengths of each node's shortest paths, in which case the nodes with the lowest distance to all other nodes is most central. This method is arguably most beneficial to apply on an information network, as it locates the nodes that can spread information to the whole network with most efficiency.

2.3.7.3 Betweenness Centrality

Betweenness centrality quantifies the number of occurrences of a node's appearance on the shortest path between two other nodes. The most central nodes by this measurement act as bridges to connect the different areas of the network.

Compares to the closeness centrality, which also utilize the shortest path property, the central nodes in betweenness measurements are more usable in many scenarios. The central nodes of closeness simply indicate they have easier access to the rest of the network than every other node; however the central nodes in betweenness act as gatekeepers for flow of information within the different regions of the network.[2] This property allows the musicians to to receive most versatile information originated from different regions that might not be connected to each other.

2.3.7.4 Eigenvector Centrality

The last centrality measurement utilizes a different strategy to find nodes of high importance. The eigenvector centrality assigns relative importance to all nodes in the network based on the concept that connection to high-scoring nodes contributes more to the score of the node in question than equal connections to low-scoring nodes. In other words a node is important when it is connected to other important nodes. A node with small number of influential contacts may outrank a node with a large number or mediocre contacts. The result indicates a certain form of authority, as it has direct connection to high influential nodes.

2.3.8 Community Detection

The connected component can be further specified into communities by dividing the densely connected areas. Each community represent the group of nodes that are more densely connected internally than with the rest of the graph. This property indicates stronger tied interaction between the particular musicians in the community.

2.3.9 Time Related Network Analysis

All the graphs treat the networks as static structures, as if a snapshot of the nodes and edges are taken at a particular moment in time. However the typical social networks are constantly altering or evolving. Single graph visualizations contain limited information for understanding the true natures behind the different networks.[2] Therefor it is useful to utilize time related dataset and observe the developments in the networks. The reasons behind the alteration of the networks can vary and the interpreted results might not offer precise explanation of particular node's development. Yet it is possible to offer appropriate descriptions of the changes based on development of network patterns and fundamental properties.

The results of the time related network comparison offers valuable information that could be utilized in researches of different disciplines within science, such as sociology, psychology and musicology. It is even possible to gain insights of the development of the music industry.

Chapter 3

Design

This chapter presents the strategy of collecting and processing data with the purpose of constructing relevant networks. These networks forms the fundamentals of the recommendation approach, which will be explained. Finally the design of the prototype is described.

3.1 Data Preparation

Certain amount of data needs to be collected and processed from the Internet to be able to create the fundamentals for the construction of networks and visualizations. This section contains the approach of data collection and data processing, including the considerations of possible data refining process. The accuracy of the data directly impacts the accuracy of the output.[9] Therefor it is highly prioritize to gather appropriate data.

3.1.1 Development Tool

The programming language *Python* has an array of practical functions and libraries for data collection and data handling. Furthermore the *NetworkX* pack-

age dedicates tools to construct and measure network properties. Therefor it is chosen development language for this project.

3.1.2 Source Of Data

First step is to acquire reliable data that contains musicians' relations to each other. It is a requirement for the data to involve a certain level of accuracy. Two possibility of the data source are considered in this project, the Rovi database - used by AllMusic, and EchoNest.

The content of the Rovi database is widely acknowledged and is created by professional data entry staff. (According to their website). All data is obtainable by the Rovi API engine.[12] This database contains vital information for the artists, such as collaborators, similar, and influencer of a certain artist.

EchoNest contains massive music-related metadata and has its own plugin for python to easily navigate through their data. It is very useful to analyse music track by track, based on their detailed metadata. Similar to Rovi database, it is very acknowledged and used by popular customers such as Spotify, MTV and BBC.[3]

The required data is focused on the individual artists, which is why the Rovi database, used by AllMusic, is most suitable and the chosen data source for this project.

3.1.3 Collect And Store Data

Each artist's data can be obtained from the Rovi API by searching for the particular artist, however searching for individual artist and combining the data is an exhaustive process to complete manually. A preferable method of obtaining a usable dataset is to construct a list of artist names and create a program to loop through each name and obtain their data from the API. For each artist the data is then stored as text format files on the same system as the one to conduct analysis on. All data files should use clear and unambiguous filenames, to avoid confusion and possible data distortion.

A considerable amount of data is required to construct a reasonable sized network, which yields optimal possibilities for interpretations. It is however not desired to fetch the data of every musician that ever existed, since it is a very time consuming and unmanageable task to complete. The expected dataset is

restricted to focus on the artists associated with the genre of Rock/Pop. This is the most exposed genre within the different media sources, which also gained undeniable quantity of the audiences. Using such a genre is to ensure the data is sufficient and to cover a large fraction of music listeners. It is unlikely for this genre to contain insufficient data from such a professional data source, in opposite to other less popular genres. The expected dataset for this project is to obtain a set of artist data within each of the different decades, which enables the option to conduct network development analysis.

3.1.4 Refine And Process Data

After the appropriate data is collected, it is then important to examine the data and determine whether the data is sufficient and usable. The data should be inspected corresponding to the artist they belong to, some artists might have different famous aliases that the API search engine responds to. The considered solution is to match the name of the data files with the exact full name of the artist, and then the duplicate files are removed. If two artists have identical full names, it might create minor data distortion to remove one of them as duplicates. However this specific scenario is presumably rare and would increase the difficulty of avoiding data distortion the relation between artists considering the ambiguous name.

If the artist list is created for each decade individually, the overall dataset can contain duplicates of artists active in different decades. It is then important to determine an appropriate data structure to store the collected data. Each decade should be assigned a folder to store the files and the combined data of all artists is filtered for duplicates and then stored into a master folder. This structure enables analysis on the dataset containing all the artists and decade specific datasets.

The raw data contains information for individual artists, and their relations. To create a network it is required to find all the relations within a specific dataset. Considering the aforementioned folder structure, it is possible to create a program to loop through the chosen folder and seek out relations of all the artists and create a list of pairs. This result is then written to a text file to be used in creation of network visualization. Each relation of the artist produces a distinct relation file, resulting in three different relation files that each corresponds to a specific network.

3.2 Constructing Networks

Different relations between artists result in different networks, in this project the focus is on the three following networks: Collaboration network, Similarity network and Influence network. It is interesting to see the similarities or disparity of the different graphs, because they originated from the same set of nodes. All three types of graphs shows relations between artists, however the relations in the different graphs contain sharply contrasting nature – collaborations are the result of conscious interactions between the artists while the musical similarity reflects the perceptions by the audience or certain authority within music (in this case, it is the experts from AllMusic). At last the influence relation could be admitted by the artists themselves or determined by the experts' analysis of the artists' works, in which case it is possible to contain combined characteristics of the two previous relations.

The goal of the network analysis is to find the potential content that might be interesting to recommend to the user. This step is the most crucial part of this project, the core data and functionality of the prototype application is built upon the results from this step.

3.2.1 Collaboration Network

Musicians often collaborate together to produce a product in the form of songs. Some musician even focuses their career on collaborating with others. In this network graph, an edge between two nodes is created based on the collaborator information from the collected data. The nature of this relation is that both musicians are equally involved in this interaction. Therefor the edges are undirected, forming an undirected graph. It is ideal to add the weight of the edges as how many songs the two musicians worked on together to further clarify the properties of the graph.

This network contains information about the collaboration behaviours of the musicians. Using the tools from previous chapter to investigate the network can reveal many insightful features of the network.

Furthermore, it is possible to determine the most effective or commercially productive collaborations by cross-reference the collaborated songs with billboard hits. The result of commercially successful collaborations and observation of the involved musicians could provide beneficial insights for the music industry.

3.2.2 Similarity Network

The different musicians can be compared to each other and some can be defined as musically similar to a certain degree. AllMusic database includes the information of which musician is similar to another musician and even has a weight score to specify the degree of similarity. This relation between musicians can be used to create a similarity network. If an artist refers to another artist as similar, a directed edge toward the referred artist is created in the graph. This property creates a directed, weighted graph, in which case the similarity references might not be reciprocal. This means some edges in the graph are asymmetric relations. The degree of nodes would have two different measures, which are the in-degree and the out-degree. Each of the values indicates different property of this node. When associated with the aspect of social network, namely the similarity network, the in-degree can be interpreted as a form of popularity or musical authority; while the out-degree can be interpreted as gregariousness or musical diversity.

This network is constructed by attributes that the experts at Rovi database has attached to the musicians, in other words the relations between musicians are not direct interactions. It is then interesting to see the difference or similarity in this artificial network's properties comparing to the collaboration network. Properties from this network have considerable potential to contribute to the recommendation system, based on the recommendation value of similar content.

3.2.3 Influence Network

Another way to explore musicians' relationships is to investigate the influence network. Musicians take inspirations from other musicians' styles or techniques, in which case a certain musical feature can be passed on from one musician to another. Once a musician inherited a certain musical trait from someone, then the person can be declared as the musician's influencer. Similar to the previous network, Rovi database include the information of influencers with a weighted score. The influencer network graph is constructed identically to the similarity network, with the directed, weighted edges. However in this case instead of popularity, it is more logical to consider the degrees of a node as authority. The musician with high in-degree has one or more traits that many other musicians inherited. This indicates that many musicians acknowledge this particular musicians' work and utilize a part of it in their own creations.

As previously mentioned the influencer network can contain characteristics of both the collaboration and the similarity networks. The properties of this net-

work are then investigated to contribute to the recommendation system.

3.2.4 Development Of The Networks

Considering the static nature of network visualizations, it is desirable to establish an understanding of the development of networks through time. The dedicated data is constructed to enable this feature. The development of the networks indicates development of social behaviour of the musicians. The results could contribute to various researches and be appealing information to present to the users of the smartphone application.

The approach is to construct the different networks based on the decade specific data. The most essential network properties are then stored to create comparison investigation. The general network properties such as the global clustering coefficient and mean geodesic length can be compared directly with each other. Other properties such as the degree distribution can be plotted into graphs and be compared within a multi-graph. Other than the mathematical properties of the networks, the most important nodes are also an interesting element to investigate.

3.3 Recommendation Based On Network Results

Once all the networks are constructed and the data from each network is analysed, the process of determining the approach of the recommendation system can begin. The goal is to define the most relevant relations between the musicians to create scopes to form recommendations on. Although the network analysis presents insightful data to the developer, they might seem cryptic for the common users of smartphones. The results from the analysis must be reconstructed into simple visualizations or descriptions that the user can easily apprehend.

When a user defines certain musician of interest, the recommendation results should revolve on this particular musician. The most obvious elements are this artist's direct relations in the different graphs: the musicians who collaborated with this artist; the musicians who have direct references of similarity to this artist; the direct influencer of this artist and even the musicians who refers to this artist as direct influence. While these direct relations form reasonable contents to be recommended to the user, this approach is somehow primitive and does not exploit the advantages of the creation of multiple networks. The

focus is to search for content that has clear relationship to the chosen musician, and create results with certain exploration value.

The tools at hand are the three networks, each presenting a different value of relations between the musicians. This leads to the investigation of which relational value is most preferable in a recommendation system. The collaboration relation indicates the musicians' direct interactions with each other. It is reasonable to assume that in general, musicians who collaborate together are likely to share a certain degree of musical similarity. However it is still possible for musicians to possess very different musical styles, even genres to collaborate together. The influence relation indicates similarity in certain traits of the musicians. However the overall style or genre of the musicians can still vary. The similarity relation, as its name suggests, is the most accurate measurement for resemblance between two musicians' overall music styles, comparing to the two former relations. For the recommendation system, the attribute of similarity is a key value. Since the similar contents to the users preferences are more likely to be appreciated by the user. If the results are exclusively formed on the similarity relations, the recommendation system resembles the content-based filtering and the capability of exploring into unfamiliar area of music is compromised. Therefor it is interesting to combine the collaboration and influence relations with the similarity relation, to create diverse results that still remain a certain level of relevance. All of the networks are constructed on the same set of database, meaning that certain part of the networks can overlap with each other. The intersections of the networks can be used to form the concepts for relevant and diverse music recommendations.

The concept is based on investigating the differences and intersections between the graphs constructed based on the user specified artist X. As shown on the figure 3.1, each area of interest are marked with the following explanations:

A - The musicians who are similar and collaborated with X

Artists in this area have direct interaction with and are perceived as musical similar to the user specified musician. This area contains artists that are very likely to fit into the user's preference.

B - The musicians who influenced X and remain similar to X

Artists in this area have benefitted and carry musical similarity with the user specified musician X. This area contains the potential predecessor of the user specified musician X.

C - The musicians who are and similar to and influenced by X

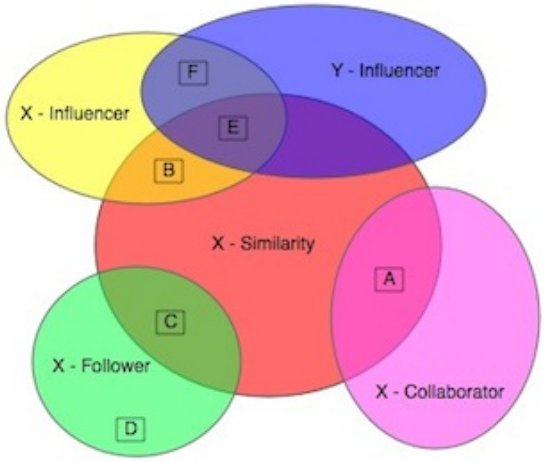


Figure 3.1: The visualiaztion of intersections between graphs as recommendation content

Artists in this area are similar to and have inherited certain musical traits from the user specified musician X. This area contains the potential successor of the user specified musician X.

D - The musicians who are influenced by X but does not possess similarity to X

Artists in this area have inherited certain musical traits from the user specified musician X but do not remain similar to this musician. This area contains the artists that are inspired by the user specified musician X and incorporated his/her musical trait into another style. This area might content music that has an unfamiliar style but still is recognizable by the user.

E - The musicians who are similar to and have common influencer with X

Artists in this area are the successors of the common influencer of the user specified musician X, while they still remain similar to this musician.

F - The musicians who have common influencer with X but do not possess similarity to X

Artists in this area inherited certain traits from the same influencer of the user specified musician X and somehow managed to exploit the inherited trait in another style of music that is not similar with the style of X. Artists in the area are different successors of the common influencer of the user specified musician X but present another approach to incorporate the same musical traits as X.

3.4 Developing The Prototype

The results from the recommendation system are designed to be presentable on a software product. The main purpose of this application is to create recommendation for the users, and then present the most interesting findings from the network analysis. The recommendation process should include certain degree of information to help the user understand the approach of the recommendation content. The most significant findings from the network analysis results should be presented to the user with simple informative text and graphs. Appropriate user interactions should be implemented to increase the desirability of the app.

3.4.1 Potential Users

The application is aimed for regular smartphone users with knowledge of basic app navigation and web browsing skills. It is also important to keep the technical terms at absolute minimum. All the informative text in the app has to be short, simple and straightforward. They serve as assistance to navigating within the app and have to be easy to understand.

The main target user group can be defined as people with a certain interest for music and the artists behind the music. Common users might be satisfied with simply being able to listen to their preferable music on the smartphone. Therefor it is required for the user to have a reasonable interest and adventure spirit to explore the recommended content.

3.4.2 Development Enviroment

As aforementioned, the combined market share of the Android and iOS smartphones is at 91%, dominating the market. Therefor is it desirable for the developer to create an application that can cross fit on both of the platforms to gain maximum potential of commercial value.

3.4.3 JQuery Mobile

One of the simplest solutions to achieve cross platform applications is to create a web-based application. JQuery Mobile is one of the many frameworks for web-based applications and is suitable for the purpose of this prototype.

JQuery Mobile is a free to use web-based framework for smartphone and tablets.[8] However using JQuery Mobile to develop a smartphone app has its advantages and disadvantages comparing to a native application.

3.4.3.1 Advantages

The most significant advantage is its ability to be adapted on both Android and iOS platforms. Which means it is not necessary to create two separate versions of code for the same application. This feature effectively reduces the resource of development and allows the developer to maintain the application without redundancy. Another convenient feature is using the browser to render the application on smartphones; it prevents the necessity to construct specified on-screen element for the varied screen size of android smartphones.

The second noteworthy advantage is the simplicity of app development using JQuery Mobile. It is much more straightforward to build than learning the interface construction of iOS or Android native apps. Testing the application can be done on the standard web browser, before implementing the application to the smartphone platform.

3.4.3.2 Disadvantages

As promising as it seems, JQuery Mobile also has its drawbacks comparing to the native applications. The first is the fact that web-based applications runs noticeably slower than the native applications. The most troublesome disadvantage is that JQuery Mobile apps do not have full access to many features of the smartphone device, such as the camera. This limits the functional possibilities for a developer. However it is not an obstacle in this project since the application does not utilize any special hardware on the device nor is it necessary to compute large amount of data.

3.4.3.3 Summary

The advantages of JQuery Mobile solve the multi-platform issue, while providing simple development environment and easy maintenance for the developer. As the drawbacks do not create critical obstacles, this framework is the chosen environment to construct the prototype on.

The final prototype should be functional on an Android smartphone.

3.4.4 Functionality Outline

The expected functionality of the prototype is based on the main purposes of the application, which can be divided into two major contents as presented below.

3.4.4.1 Network Based Recommendations

First of all, the user must be able to choose an artist of interest within the existing dataset. When an artist is chosen, the application should show the essential information of this artist, such as names, picture and a short description. The recommendation approach is then applied on the artist and creates the output content. Then the recommendation results are presented to the user, preferably with a short audio preview.

3.4.4.2 Presentation Of Network Analysis Results

Results from the network analysis are constructed in a clear simple presentation to show to the user. This is considered an informative page, similar to an article in a web browser. This function does not require specific user interactions. However it is crucial to describe the results as simple as possible.

The artists with certain significance within the network results can be used as suggestions to the user.

3.4.5 Structural Mock-Up

The application allows navigation through four different pages:

- Home page
- Artist List page
- Artist Profile page
- Result Presentation page

3.4.5.1 General Thoughts

The general idea is to briefly inform the two major contents of functionality outline on the Home page and use buttons to navigate to each of the contents.

The network based recommendation requires the user to choose an artist to construct the recommendation content upon. Therefor the first step is to choose an artist from the Artist List page, which contains all the available artists in the dataset. When the user has chosen a specific artist, this artist's Artist Profile page is shown and presents this artist's basic information while the recommendation content from different approaches are presented.

The second button on Home page navigates the user to the Result Presentation page, where the most significant findings from the network analysis are shown with informative text and simple graph visualizations.

Based on these descriptions, the Artist Profile page is the most important component of this application. The most vital functionality and the results of the recommendation system are shown on this page. Therefor it requires most attention to design.

3.4.5.2 Artist Profile

Since the recommendation is based on the networks, it is logical to assume the result could be shown as a small scope of the network. The first mock up is designed as shown in A.1.

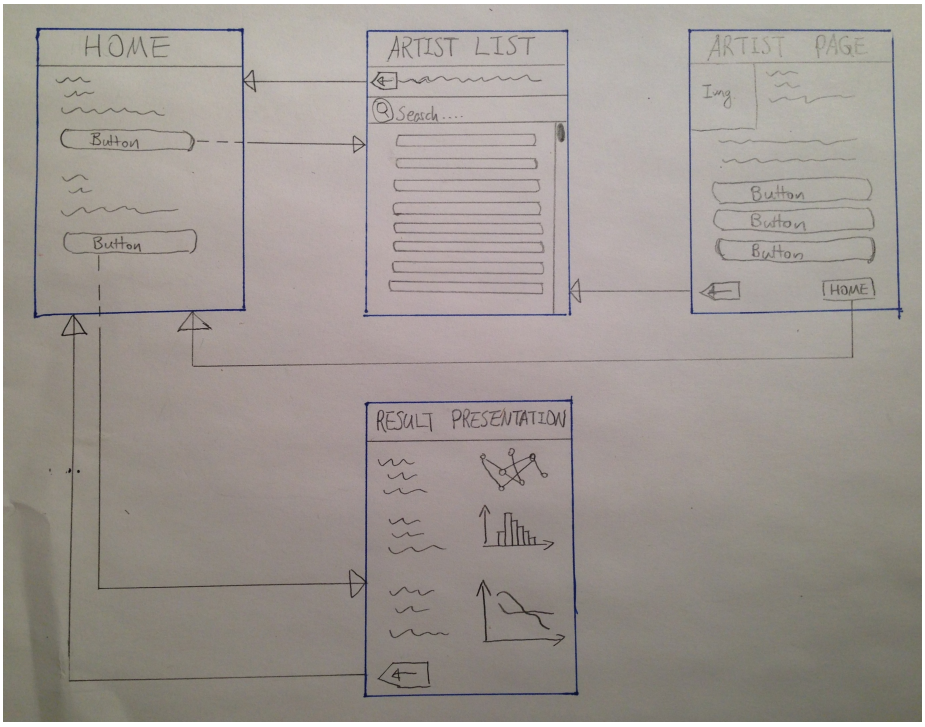
However this design does not verify the clear indications and have very complex user interactions. Common users would not be able to immediately understand the ideas behind the network. This is proven by testing this design with a potential user and gained immediate response to confirm this assumption.

A simpler, more manageable design is utilized to avoid this issue. The new design simply categorizes the recommendation results by the approach and presents

them as a collapsible button to maintain simplicity. The second design mock up can be viewed in A.2

3.4.5.3 Navigation Process

The overall mock up for the navigation process within the prototype can be viewed below:



Chapter 4

Implementation

This chapter reveals the detailed process of data collection and data processing. The in-depth analysis of the network results are also presented as well as the implementation of the recommendation approach. Finally the implementation according to the design of the prototype is explained.

4.1 Collecting And Organising Data

Utilizing the Rovi API engine, which is dedicated for AllMusic database, the required data is collected. This engine includes many practical operations, the most noteworthy for this project is the *Search* and *Autocomplete*.

The *Search* call takes artist names as inputs and shows JSON data as output. The output data can be specified to include practical attributes for this project. The most suitable attributes of each artist for this project are listed as shown in Appendix 2 B.1.

The parameter *include=all* is used to ensure the completeness of the data, which also avoids redundant call operations to further acquire data.

The first obstacle to be encountered is the creation of artist list. It is preferable to acquire data for the most relevant artists who are active in specific decades. The solution is to utilize the *Autocomplete* call from the Rovi API. The *Autocomplete* command includes parameters such as query and filter to complete this task.

The query parameter can be used to limit the result starting with a certain initial. Including the parameter *query=a* in the API call returns musicians with the initial letter a. The filter parameter further specifies the output by different aspects such as genre and decade. These parameters are combined into the API call to return results containing the most relevant Pop/Rock artists in every specified decade.

However the Rovi API has a call limit for each developer. The limit is 3500 API calls per 24 hours. The approach for data collection is then to use *Autocomplete* command that limits results of each initial to 150, which returns a list that contains maximum 3900 artists for every decade. However the list may contain duplicates, such as "The Eagles" responds to the initials T and E. After removing the duplicates, the length of data is approximately 3000 for every decade.

When the lists are completed, it is possible to loop through them and utilize the Search call on every artist to create the dataset.

4.2 Data Processing

It is necessary to extract the relation data to construct networks from the artist database. But before initiating this process, the data has to be refined to avoid as much data distortion as possible. An important fact is that the relation attributes only refer to other musicians by their full name.

4.2.1 Sufficiency And Redundancy

First of all it is important to investigate whether the data contains a level of sufficiency. The collection process does not inspect the API call result; every output is simply stored into text files. An artist can exist in the Rovi API database; however it is not guaranteed that the information of this artist is sufficient. In some cases the output contains nothing else than the artist name, leaving all the other attributes as blank. These files are not sufficient enough to contribute to this project, and therefor are removed from the database.

When facing a database with a large amount of data, it is then vital to check for redundancy. Some artist are recognizable by their famous alias, an example of this is “Art Tatum”, who is simply famously recognized as “Art”. This also means that the API call responds to both names with the same data output. The collection process then stores both output as separate files, creating redundant data. Based on this scenario, all the data is then renamed corresponding to the full name of the artist they belong to, which diminishes the chance of duplicate data content.

4.2.2 Band Filtering

The final step of refining the dataset is to determine the degree of usefulness of the data. This project is based on creating individual artist's social networks. The database however contains information of both artists and bands. In this specific setting, the data that describes a full band is not as useful as the individual artists. The dataset is then filtered to contain only single artist data. Due to the possibility that band data can benefit in certain scenarios or further research subjects, they are not removed but simply kept in a separate folder.

The finished dataset has the following number of artists in each folder:

	1960	1970	1980	1990	2000	2010	Combined
All	2866	3091	3665	3643	3681	2836	10596
Single artists	1593	1756	1874	1550	1440	1213	4769

From the original artist-list, approximately 40-50% of the list contains band names instead of single artists.

After the refining process, the dataset is ready to be processed into the required materials for network construction. The data structure assigns a separate folder to contain artists for each decade and a folder for the all the artists combined. A program is then constructed to loop through the artist files individually and seek out the different relations of this artist. The relational data is stored temporarily as a dictionary in the program. Each of the relations is then inspected for filtering. The first inspection is whether the referred artist exists in the collected dataset. If an artist has a connection with another artist who does not exist in the dataset, then this connection is filtered. The idea is to construct the network that shows relations between musicians within the existing dataset. The second inspection is to determine the correctness of the relations for each of the networks. Each relation represents an edge between vertices in the network graph and an edge contains different properties depending on the type of the

graph. In an undirected graph such as the collaboration network, an edge from vertex A to vertex B is identical with the edge from B to A. However these edges are added with a direction, which indicate their difference in a directed graph such as the similarity or influence networks. Based on the undirected property in collaboration networks, the edge between two artists is checked for duplicates and only one edge is remained between any two nodes. Furthermore the similarity and influence networks add a weighted score on every edge.

The output of the relation is stored as text format and is structured as the examples below:

Collaboration:

ArtistA (tab) ArtistB (newline)

Similarity: (Influence follows the same structure)

ArtistA (tab) ArtistB (tab) weight (newline)

4.3 Network Presentation And Results

The networks are constructed and studied by using the *NetworkX* package for *Python*. *NetworkX* enables the creation and visualization of complex networks and includes functions to calculate the different network properties. The visualizations are shown in the Appendix C. This section contains the in depth analysis of the different networks and put the different networks in comparison to find as much information as possible for understanding the similarity and differences of them.

4.3.1 Graph Construction And Process Of Data Calculation

It is simple to construct the collaboration network, based on its undirected and unweighted property. The graph G is simply constructed by adding the nodes and edges from the relational files.

The construction of the similarity and influence networks is proven to be trickier than the previous one. The approach for the similarity network is to reconstruct the directed and weighted graph G to an undirected weighted graph Gu, by

only retaining the reciprocal relations between two nodes. This is accomplished by using the *(to directed)* function of *NetworkX* with the parameter: *reciprocal=True*. In the similarity network, it is logical to consider that the most beneficial relations for the recommendation system are to investigate the artists with a mutual reference to each other as similar. It is also logical to assume that the Rovi database would assign both of the references the same weight, considering the mutual reference.

The direction and weight of the edges are much more important in the influence network, which causes problems in calculating outputs. Many of the property calculations can only be conducted on an undirected and unweighted graph, such as the clustering, assortativity and centrality. Simply converting the graph to an undirected graph is somewhat inconsiderate and affects the accuracy of the ouput to a certain level. However the outputs are still containing a certain value for comparison and interpretations. In this case, the degree relations such as the highest in- and out-degree are calculated on the original, directed graph. The rest of the values are calculated on the converted, undirected graph. The conversion is similar to the similarity network, with the only exception of the reciprocal value is false in this case, as it is very unlikely for two musicians to refer to each other as musical influencers.

The next step is to extract the giant component, because some of the calculations and properties cannot be applied on multiple isolated components. When the giant component is extracted, it is crucial to observe whether the data contains the approximate level of sufficiency as before the extraction. If a large complex component is removed, the calculation output will not represent the true nature of this network, and the interpretations will be inaccurate.

4.3.2 General Graph Results

The first step of interpreting the graph is to do a direct comparison of the different types of networks. In this step the focus is on the difference and similarity of the network types in this project. The networks are constructed on the combined dataset containing artists from all the decades. The significant disparity is observed from the table below:

Property	Collaboration Network	Similarity Network	Influence Network
n	1261	2290	1436
m	1951	8506	4510
GC size	907	2172	1395
GC edges	1688	8426	4488
Com	28	16	11
d/d_{max}	6.7 / 21	4.7 / 19	3.8 / 8
C	0.24	0.18	0.06
R	0.09	0.16	-0.11
k_{max}	Jim Keltner(48)	Eric Clapton (69)	Bob Dylan (248)
B.Cen	Jim Keltner (0.29)	David Bowie (0.04)	Bob Dylan (0.33)
E.Cen	Jim Keltner (0.32)	James Taylor (0.24)	Bob Dylan (0.50)

4.3.2.1 Network Size Comparison

The full dataset contains 4769 single artists. The networks contain various number of nodes n. The collaboration network contains approximately 26% of the full dataset, creating the smallest network of all three. The influence network is slightly larger by containing 30% of the full dataset while similarity is proven to be the largest network, by including 48%.

Although the collaboration network and the influence network are similar in size, the influence network contains more than double the number of edges. The average edge per node is only 1.5 for the collaboration network, while this value is 3.1 for the influence network and 3.7 for the similarity network. The assumption of the collaboration network is the least connected or complex network of the three can be based on these values.

4.3.2.2 Giant Component Comparison

The results showcase the consequences of extracting the giant component.

	Collaboration	Similar	Influence
Node difference	354 (28%)	118 (5%)	41 (3%)
Edge difference	263 (13%)	80 (0.9%)	22 (0.5%)
Edge per node	0.74	0.68	0.53

The portions being subtracted are most noteworthy in the collaboration network, while the network is reduced by 28%. However the low value of edge per

node indicates that these components are either weakly connected or contains many isolated pairs. The removed network components are relatively small in the other two networks while the edge per node value remains small.

The network construction method diminishes any chances of existence of isolated nodes, which indicates the smallest possible network component is an isolate pair of nodes. This property restricts the edge per node value to be equal or above 0.5. The value from the removed segment in the influence network is very close to this limit, which indicates the high existence of isolated pairs with a very few open triplet.

4.3.2.3 Number Of Communities

The comparison graph shows significant differences of the number of communities within the networks. The collaboration network contains highest community amount, while being the smallest network in size. Each community is an internally strongly connected segment of the component. The large amount of communities indicate the network contains many internally strongly tied segments while the interaction between the segments are weakly tied. This observation supports the earlier assumption of the collaboration network is the weakest connected network of the three. The removed segment by extracting the giant component has very low impact on the similarity and influence networks. While their low edge per node value further indicate the low consequence of removing these sections on the overall interpretation of the networks.

However the collaboration network has reduced almost a third of its size, therefor it requires another step to investigate the consequences. The approach is to compare the number of communities before and after the extraction of giant component. The network contains 159 communities before the extraction, while remaining 28 within the giant component. There are 131 communities within the removed section, which contains 354 nodes. With this information it is possible to calculate simple values to assist on understanding the removed section. There are in average 2.7 nodes and 2 edges within each community. This attribute can be roughly visualized into an isolated open triplet as each removed community. The significance of these communities to the overall output data is negligible. Therefor the extraction of the giant component does not present remarkable consequences for overall interpretation.

4.3.2.4 Geodesic Comparison

The longest geodesic length defines the diameter of the networks. Despite the large network size, all the mean geodesic length are below 7, and the diameter of the networks are no larger than 21. The mean geodesic length confirms the small-world phenomenon, showing that the average distance between two nodes in the network is approximately 6.[2] However the mean geodesic lengths vary in a board spectrum while the collaboration possesses the closest value to this specific phenomenon. One interesting observation is the results of the influencer network. This network has a surprisingly low diameter and mean geodesic length. This will be investigated by the centrality measurements that utilize the geodesic length.

4.3.2.5 Clustering Comparison

The global clustering coefficient, also known as the transitivity ratio, is an indication of how tightly knit a network is.[2, 15] The collaboration and similarity shows a much higher value than the influence network. To understand this difference, the fundamental natures of these networks must be considered. The transitivity ratio can be described as the probability of a connection between two randomly selected neighbours of a certain node. It is reasonable to assume that two musicians are more likely to consider a collaboration if they both worked with a certain musician. Furthermore it is also logical to consider two musicians to be similar, if they have a certain musician they both are similar. Following these arguments the collaboration and similarity network have a transitivity value close to 20%, which can be considered as reasonable clustering. The influencer graph however contains a contrasting nature. The reference of a common influencer by two musicians does not necessarily indicate they have any relations or interactions with each other. Even when the graph is converted to undirected, it still has higher requirements than the other networks to form a closed triplet. Such a triplet is formed by a very specific scenario: if one of A or B is the influencer of the other and both A and B is influencers for C.

The influence network has certain resemblance to the layer-structured hierarchical tree. The relations in this network represent a very specific type of relations and therefor some attributes differ from the typical social networks.

4.3.2.6 Most Central Nodes

Another measurement of understanding the networks is to locate the most important nodes within the network. However the importance of a node is measure in different aspects, each demonstrating the relational importance of a set of nodes. The interpretation is based on the overview gained by utilizing different measurements.

The first measurement is to find the nodes with the highest degree, or direct neighbours. This measurement finds the nodes that can directly impact largest fractions of the whole network. The similarity network is converted into undirected graph, which can be measured using the same method as for collaboration network. The nodes with the highest edges directly attached are the most central nodes. The most central node in the collaboration network is Jim Keltner, with a degree of 48, while his counterpart in the similarity network is Eric Clapton with 69 degrees. As a successful session drummer known to work together with three of the members from The Beatles, it is reasonable to think that he has collaborated with most musicians in this dataset. As for similarity, it is as expected that the most central node would be a familiar name. According to the popularity assumption in the analysis chapter, it is not unlikely that the most referred artist in similarity is the legendary musician whose name is associated with the famous graffiti "Clapton is God" during the golden age of rock. The measurement is slightly different for the influence network, given the significance of the directed relations. The centrality splits into two different aspects – the in-degree and out-degree centrality. The highest in-degree node is Bob Dylan, with 245 in-degrees. It is reasonable to consider him as an authority figure, as 17% of the musicians in the network refer directly to him as influencer. On the opposite, the musician with highest out-degree is Slash, the memorable guitarist from Guns'N'Roses, arguably one of the most popular bands in the 1990s. Slash has 18 out-degrees, which indicates him as the most adaptable musician within this dataset.

However a node might be important for the whole network without having the highest degrees. The betweeness centrality is based on the the nodes occurrence on all the shortest paths. These nodes acts as bridges to the different components of the network, also functions as gatekeeper for the flow of information. To interpret these nodes further, their properties are inspected and compared.

Highest betweenness centrality nodes:

	Degree(rank)	Deg Centrality	Local clustering
Col - Jim Keltner	48(1)	0.05	0.063
Sim - David Bowie	70(5)	0.02	0.065
Inf - Bob Dylan	248(1)	0.18	0.020

The local clustering for top betweenness nodes in collaboration network and similarity network shows resemblance while this value is much lower for the influence network. These local clustering values are generally very low, indicating the low tendency of these nodes' neighbour to connect to each other. This confirms the theory that these nodes act as bridge between different, unconnected regions within the network. On the other hand, it is also important to observe that these nodes have the very high rank in the degree centrality. This could indicate that the nodes with many neighbour who does not tend to connect to each other, are the ones acting as bridge. For collaboration network, this means the most central node would collaborate with many other artists who does not have collaboration relation together. This quality suits the description of an session musician such as Jim Keltner. It is possible to imagine that people who does not collaborate together could have a unique style, approach or idea on creating music, which Jim Keltner have the first hand access to. This could increase his creativity by being exposed to musical variation. As for the central nodes in the similarity in network, this property can be considered as the musical diversity of the musician, since many musically different artists refer to this musician as commonly similar. At last the central musician in the influence network requires further specification of the follower to be interpreted with certain accuracy. The number of followers and their active period compares to this musician can indicate whether this musician is an authority of his own period or he has inspired a movement, style or even a generation of musicians.

The last measurement is the eigenvector centrality, which can be considered roughly to reveal the nodes that have highest connections with other significant nodes. This measurement reveals Jim Keltner and Bob Dylan as the most central nodes within the collaboration and influence networks respectively. Considering that Jim Keltner has collaboration relations with some of the most acknowledged musicians and Bob Dylan has direct relation with 17% of the network, the results seem reasonable. This measurement reveals a new name – James Taylor in the similarity networks. As an owner of 5 Grammy awards and the 84th place holder on the "100 Greatest artists of all time", it is not difficult to imagine the authority of this artist and the probability of him possessing direct relations with other extraordinary musicians.

4.3.3 Development Analysis

The time specified dataset allows the observation of network development and opportunity to further interpret the difference and similarity between the networks.

First of all it is important to gain an overview of the data, this can be done by determine the difference in the data. The difference is calculated by checking how many artists exist in the previous decade.

	1970	1980	1990	2000	2010
Files	43,1%	53,5%	59,8%	64,5%	79,3%
Collaboration	46,4%	63,4%	88,7%	95,5%	59,6%
Influence	56,9%	62,5%	74,1%	74,9%	90,2%
Similarity	50,7%	62,2%	69,2%	69,8%	85,5%

Musicians often have an active career spanning over several decades. The file difference indicates that there are 43,1% artists who are active in the 1960s and also in the 1970s. The growth of the data in the 1970 is then only containing 66,9% new artists. This growth diminishes to only 20,7% as seen in the 2010s difference value. This information directly affects the growth of the other calculated properties. The collaboration difference indicates how many artists in the current decade also exist in the collaboration network in the previous decade. This roughly shows the expansion of the collaboration network. It seems that the new artists in the 1990s and 2000s do not work together as much as the other decades. It is a possibility that the new artists collaborates with the ones that exist in the previous decades instead of internally collaborating together. This diminishing growth is also observed in both of the other networks. However the collaboration network shows an irregular growth comparing to all other networks. It has the most significant reduction of growth in the 1990s and 2000s and is the only network that regains growth of new artists in the 2010s. The reduction of growth within the influence and similarity netwroks can be explained by the relational nature, it is reasonable to assume that artists would have a higher tendency to be influenced by previously existing artists and therefore be considered similar to them.

Finally it is also important to relate this finding to the constructed dataset. The dataset is collected by the most popular artists, which means the growth shows whether the popular musicians of each decade is dominated by existing artists or new artists.

The next step is to calculate the network properties on all the networks in all

the decades. The result is shown in the table in Appendix D.1.

First observation is the size of the giant component which resembles the previous network comparison. The similarity and influence network have a very dominant giant component while the collaboration networks giant component size varies in a board spectrum. The most noteworthy is the collaboration network for the 1980s. This network has reduced 40% in nodes but maintains almost 94% of its edges. This indicates that these remaining nodes are very tightly knitted together. This is also confirmed by the highest global clustering coefficient, indicating that the musicians have the highest tendency to collaborate with other musicians with common collaborator. The ratio between collaboration network size and data size shows a growth of collaboration within musicians from the 1960s to the 1970s and constantly decreases from 1970s to the lowest 15,9% in the 2010s. In the 2010s, only 193 out of 1213 artists work together and 83% of these musicians tend to create collaboration isolated from other collaborating musicians. The global clustering coefficient further indicates the reduced connectivity within the giant component. This indicates the decreasing collaboration between the active musicians in the decade. The collaboration network is also the only network that reduces in the artist to file ratio. Both influence and similarity network have very insignificant variation within the network properties such as mean geodesic length, global clustering and assortativity. However they both show constant growth of proportion comparing to the amount of artists. These properties combined indicate the uniform growth of these two networks, they grow larger by proportion of total artists but the network structure remains similar.

4.3.4 Artificial Networks vs Realworld Networks

Summing up the differences observed by property comparison, it is possible to discuss the distinct features of an artificial network and a self-organised social network.

The collaboration is the only self-organised social network within this comparison and this graph shows significant differences to the other networks. This network contains the highest amount of isolated components. Almost 30% of the network is formed by small isolated collaborations. As opposite to the other two networks with 95% of the network represented within in the giant component. The most significant difference is the network structure. The collaboration network has the lowest average degree, indicating a lower connected network, however the global clustering coefficient proves that this network is more tightly knitted than the others. The collaboration network also contains nearly twice as many communities as the other networks. Combining these two features, it

is possible to assume that this network is scattered into many internally tightly knitted communities while these communities are not densely connected to each other. Considering the small-world phenomenon, the collaboration network is also the closest one to a real-world network by the mean geodesic measurement.

The similarity network shows certain resemblance to the collaboration network, but it also contains a smaller mean geodesic length considering the similar network diameter. An area containing a large fraction of the nodes with very small shortest paths between nodes, which indicate a very high density and connectivity, could cause this. The smaller number of community and lower value of global clustering could further confirm this assumption. Combining these features, it is possible to consider that this network has a high-density centre, containing several super nodes that are connected to a large fraction of this network. This network is designed by human perceptions, filtered by subjective opinion, thus the difference comparing to the self-organised network.

The influence network was assumed to contain elements of both networks. This assumption is contradicted by the network data. Instead of combining the attributes, its properties lean towards a more exaggerated version of similarity network. As earlier mentioned the network harbour resemblance to hierarchical trees rather than social networks. This assumption is confirmed to a certain degree by the output data. The remarkable low mean geodesic length, global clustering and assortativity values indicate this networks fundamental differences to the two previous ones. A small amount of super nodes forms the centre mass of this network. The degree sum of top ten degree-central nodes is 995. If their interconnections are looked aside, this sum forms 71% of the nodes in the entire network. These super nodes have considerable capability of reaching to a large fraction of the network, which explains the low mean geodesic length. The inheriting nature of the relations divides the nodes into roles of influencer and follower, which contradict the assortativity calculation of connection tendency between similar nodes, resulting the negative assortativity value.

4.3.5 Succesful Collaborations

To further specify the collaboration and help interpreting the most significant collaborations, the network is added with an extra property. The goal is to seek out successful collaborations that have resulted billboard placement of songs. The approach is to seek out all the songs of all collaboration and cross-reference the song names to the billboard data. The collected artist data contains all the songs they created. It is then possible to seek out songs with same name and same ID to find the collaborated songs.

The graph constructed by this dataset contains almost exclusively isolated pairs of nodes, as shown in Appendix E.1.

This could be caused by the lack of song data for many artists, and assumedly also incomplete song data of each artist. The artist "Jim Keltner", as the artist with most collaboration neighbours, has only 6 songs from the AllMusic database.

This finding is rather unexpected considering the data source. However is it still possible to proceed the approach and find the most successful collaboration within this limited dataset.

A new data source is required to collect the billboard information. The most ideal database would be the Billboard API (www.billboard.com). However this API is no longer offering a public API service. Therefor it is necessary to seek out alternatives. The chosen data source is a website called TsorT (http://tsort.info/music/). This web site collects the US billboard information for every year between 1920 and 2009. This is chosen to be the cross-reference data. This data source is arguably less accurate than the Billboard API. However it contains a large set of data which they claim to have collected through reliable source. With such a data source, it is necessary to be thorough when handling the data. The approach is to collect the billboard top 100 hits from every year between 1960 and 2009. This means the data of billboard hits in 2010s is missing. The result is 4610 distinct songs and they're used to compare with the data collected from the processed data.

Output are shown in Appendix E.2.

From the output it is clear that the data has reduced size once again. It is not necessary to plot this output to see that it will not be a usable visualization, as it follows the structure of exclusively containing isolated pairs. By observing this output, the most successful collaboration is between Chris Spedding and Robert Gordon, with 11 billboard hits. However this collaboration is not the most efficient within this data. The song amount per billboard hits ratio is at 20%, while the highest efficiency is the collaboration between Leon Russel and Willie Nelson, with 25%. The second highest is between Colin Bluntstone and Rod Argent with 23%.

4.3.6 Recommendation Data

The recommendation data follows the intersection observation design as mentioned in the design chapter. The approach is to not visualize every artist, but

to use their relational data to find intersections and differences. For each artist, all the corresponding relational data is collected, processed and stored. For example, the artist Eric Clapton would have three different relational data in total, each defining his direct neighbours in the different networks. The approaches are described corresponding to the marked graph section in (From graph in design).

A

Result is computed based on the intersection of direct neighbours in similarity and collaboration networks. The output is sorted by similarity weight.

B

Result is computed on intersection of direct neighbours in similarity network and the out-degrees of influence network. Both relations are weighted and the final data is sorted by the sum of both weights.

C

Result is computed on intersection of direct neighbours in similarity network and the in-degrees of influence network. Both relations are weighted and the final data is sorted by the sum of both weights.

D

Result is computed by taking the difference between the in-degrees of influence network and the direct neighbours of similarity network. Results show the followers who is not related to this artist in the similarity network. The final output is sorted by the weight of influence relation.

E

Result is computed on intersection between direct neighbours in similarity network and artists with common influencer. Both relations are weighted therefor the output is sorted by the sum of both weights.

F

Result is computed by taking the difference between artists with common influencer and the direct neighbours in similarity network. The final output is sorted by the weight of these artists' relation to the common influencers.

The approach E and F are computed by finding all the influencers by the user specified artist and find all their direct followers. This set of data is named artists with common influencer.

The results computed for Eric Clapton is shown in the Appendix F.1

Ten highest valued data is presented as output from this program. If the method contains more than ten results, the ten output is chosen by randomly select the ones with highest output.

4.4 JQuery Mobile Implementation

The required data is constructed to be implemented into the prototype. This section focuses on creating the prototype following the mock-ups design.

4.4.1 General Features

The general structure of this application contains 4 simple pages, following the design mock-up. This app primarily focus on presenting constructed data rather than process data and construct output based on the user input. Therefor the components are either for informative or navigation purposes. The fundamental components within these pages are textfields, buttons and lists. Certain more advanced features such as audio preview are also included .

The Home page contains a short description of the app, simple instructions of how to use it and two buttons directing to the two major functionalities of this prototype – Artist recommendation and Result presentation.

The page containing all artists as a list bridges the Home page and the specific Artist Profile page. This page present all the existing artists with a simple filter/search bar. The Artist Profile page includes a brief description of this artist and the recommendations are presented in the corresponding categories which they are formed upon. Each category is a collapsible tab and contains 2 results per method for maintaining the simplistic overview. Each result has an 30 sec audio preview by accessing the Wimp server.[16]

The Result Presentation page contains the most important findings of the network analysis. The data is primarily shown as text, certain images of graphs can be included to visualize the findings. All of the mentioned artists in this page should be assigned as buttons to direct to the corresponding artist profile.

4.4.2 Data Accessibility

This application does not contain a certain database to fetch information from. All the presented data are computed and organized during the network analysis process using *Python*. Each of the pages in the application is simply generated by *Python* according to the *Html* code structure. The advantage is the increased stability – errors can only occur if there are reference mistakes within the *Html* code. Furthermore, the user does not need internet access to view results, with the only exception of audio preview.

However if the results are based on processing user input and access data from an internet server, it is possible to expand the data on the server instead of upgrading the app. It will not be necessary for the user to update the application to access new data. This reduces the developer's maintenance workload and avoids going through the complicated application validation process of publishing an update. The only data accessed using the internet is the audio preview. This data is accessed from Wimp server.

4.5 Smartphone Implemenation

The advantage of JQuery Mobile's simple smartphone implementation is revealed by the open source framework PhoneGap.[11] Both implementation to Android and iOS is done by following the "Getting Started Guides" on their website.[4] This guide generates a sample project for the Android SDK or iOS SDK. Then the html code constructed in JQuery Mobile simply has to replace the content of the *www* folder within the sample projects. The prototype is then available to be run on the simulator of both Android and iOS systems.

The prototype is easily installed on an actual Android device for testing purposes. However it is not as simple for the iOS system, because apple requires the license of the iOS Developer Program to implement the prototype on an actual iOS device. The price is $99 per year for being a member of this program. Therefor the prototype is only functional within the iOS simulator. This is sufficient to test the basic functionalities.

Chapter 5

Evaluation

This chapter focuses on validating the prototype and the actual market. The functionalities are tested by constructing use cases. All the user feedbacks are gathered and dicussed.

5.1 Prototype Completeness

The current prototype contains one complete Artist Profile for the musician - Eric Clapton. This is the first iteration of the database, the rest of the artist pages are construct with identical approach and is expected to be completed on the presentation date. The Result Presentation page contains the most significant findings presented in text format, this page will also be extended and refined on further findings.

5.2 Prototype Robustness

There is no specific user input other than the simple search bar in the artist database. It simply filters the shown list of artists by comparing the input.

Every other operation in the application is based on connecting another page by clicking on a button. This minimizes the possibility of program crash or dysfunction based on the user's interactions. This structure ensures the robustness of the application assuming the code is functional. However the Artist List search bar could include a suggestion function to assist the user, in the case if the user forgot the full name of the artist of interest.

The simple cross platform fitting and the simple development environment is both appreciated in this project, however testing phase have also showed performance issues based on the disadvantages of using JQuery Mobile. It is incredibly simple to develop an application interface using this framework and implement the web-based application on a chosen smartphone platform. But the lowered performance is sensed when using the Artist List page. This list contains a total of 4769 artists from the database as list items, and by loading the entire list on opening the page cause a delay. This can be improved by hiding the list before user input in the search bar.

5.3 Use Cases

The use cases test the two main purposes of this application for their functionality. Use case 1 is focused on artist based recommendations and use case 2 is focused on viewing the analysis results.

5.3.1 Use Case 1

The user is interested in an artist, from the recommendation the user become curious about a new musician from the recommendations and performs further navigation.

Prerequisites:

The user has an artist of interest and knows his full or partial name and the artist exists in the database.

Main Scenario:

1. The user clicks on the Artist recommendation button from Home page

2. The user inputs the name of the artist of interest and clicks on his/her name
3. The Artist page shows the profile and recommendation results
4. The user become interested in another artist based on the audio preview and clicks on his name
5. Repeat from 3

5.3.2 Use Case 2

The user is interested in viewing the analysis results and finds a musician of interest from the analysis result.

Main Scenario:

1. The user clicks on Network Insights button from the Home page
2. The user views the result data and become interested on a certain musician
3. The user use the back button to go back to the Home page
4. The user clicks on Artist recommendation button from Home page
5. The user finds the name of the artist list and views this artist's profile

5.4 User Feedback

The user feedbacks give the developer an overview of the potential of the existing market value and how well the design lives up to its expectations in practice. This project includes a simplified version of the thorough user feedback analysis due to the insufficient resources. The approach is to ask the users a list of simple questions and let the users test the current prototype based on the use cases. The whole process is done electronically, contradicting the usual personal interview. Since the application is web-based, the prototype can be sent as a compressed file together with the questions. This approach is not as complete as the preferable method. Nevertheless it collected valuable information comparing to the limited method. The users are within the age of 18-23. All of them own and use a smartphone on daily basis. The whole process of the simplified user interview is limited to 15 minutes.

A total of 6 users participates in the testing process and provided feedback. The questions and corresponding feedback can be viewed in the Appendix G.1 and Appendix G.2.

5.4.1 Validating the market and the network approach

The questions in the first part of the feedback test are based on validating the market.

The answers are collected and compared to create an overview. Most users are not particularly familiar with the recommendation systems, at least not the specific approaches used in the different applications. This confirms the earlier assumption of most existing recommendation systems does not inform the user why and how the content is created. But the more vital insight is that most of the users would like to know the general idea behind an approach of recommendation. Thus this functionality is desired and absent in the market. This validates the market potential and user potential behind this functionality.

While the functionality is validated, the next question is to determine whether the network recommendation approach fulfils this functionality.

The last four questions are focus on validating whether the network approach meets the demand of the users. Most of the users prefer recommendations mainly containing the similar content but would also appreciate different content to be able to explore into new musical styles. But if the results are different from what they listen to, the majority prefer the content to be based on a certain criteria. These criteria could be the musician networks as all the users are positively interested in. All these information combined validates the potential usefulness of the results from this approach.

5.4.2 User Experience And Findings

In this project, the approach is a simplified version of the interview – to acquire user feedback and general experience impression. The prototype has been sent out to the potential users with the two tasks. The tasks are based on the use cases:

- Find the musician: Eric Clapton, and acquire one artist from the recommendation

- Find who has inspired most musicians

The general impression was that the program, within its currently limited area, is robust in the sense that the program does not crash and the user does not get stuck during the navigation. The first task was completed without difficulty. The second task took longer for the users to complete. This complication is mainly based on the currently primitive design layout for the Result Presentation page.

Positive Feedbacks

- Nice, simple and clean design
- The audio preview is useful
- Good short description for each artist(Headline bio from AllMusic)
- The navigation system is simple and manageable to use
- The recommendation results seem reasonable
- Most users understand the reason behind each recommendation content
- Most users find the results interesting or useful

Improvement Areas And Suggestions

1. Result page is confusing and messy
2. Too much white space on home page
3. Loading time on artist list is too long
4. Search artist could be placed on the front page instead
5. Back button on the artist list page is too far down
6. More songs from the recommended artists
7. More results within the same recommendation category
8. Recommendation tabs could be labelled simpler
9. Include visualizations to allow exploring the network
10. Artist profile could be extended

11. Artist profile should contain top songs of this particular artist

12. Favourite function or navigation history to keep track of the visited musicians

The most noteworthy positive feedbacks are the users' reaction to the recommendation results. Most of the users understand the basic approach to form the results and find the results reasonable, interesting and useful. This validates the core value of the applications main purpose. Both the navigation and design is developed as simple as possible are approved by the users.

The first two improvement areas are based on the layout of the application. These are the easiest to fix and therefor seen as minor issues. However they grant valuable insights on how to design the next mock-up.

The problems 3-5 are based on the Artist list. Almost every user reports a long delay on loading this list. To improve this page, the suggestion can be followed to a certain extend. The Artist List should be hidden before the user interacts with the search bar and the Back navigation button should be at the top of this page instead. The idea of implementing the Search bar on the Home page instead is viable and should be considered in future design.

The problems 4-11 are based on extending the Artist Profile. The prototype contains only the very basic information of this artist and it is expected for the users to suggest which information they would like to be added. All the suggestions will be considered to be implemented in the future version of this prototype.

The last suggestion reveals a whole new functionality within the application. If it is possible to navigate through different musicians, it makes perfect sense to save certain musician of particular interest. This functionality will be considered to add to the future version of this prototype

5.4.3 On Device Testing

The prototype is integrated to an Android smartphone to test its functionalities. The screen size is automatically adjusted by the JQuery Mobile framework. However the loading process on the Artist List is too slow. The smartphone used is an old device, with a reasonably slow running time and low CPU power. The processing of showing the Artist List page is approximately 10 minutes. This issue has evolved from a minor issue from the user feedback test to a critical

problem for the actual prototype. The possible solutions will be presented in the discussion chapter.

Chapter 6

Discussion

This chapter reflects the project on the accomplishments. The possible errors and improvements are presented and dicussed as well as the possible subjects for future work.

6.1 What Has Been Accomplished

Firstly the collected data is proven to be sufficient to construct and analyse the different networks required for the approach of the recommendations. The interpretation of the different networks revealed the difference between self-organised networks and artificial networks, as well as presenting certain nodes of interests for further analysis methods. Each of the most interesting network intersections and differences are inspected and utilized to create recommendation results.

A functional prototype is developed and integrated to both Android and iOS platforms to present the results. The market value and usability is tested and confirmed by the simplified user experience. The prototype is tested with use cases to confirm its expected functionalities.

6.2 Possible Data Distortion And Thoughts

The output from the successful collaboration seems to be caused by data distortion. The data of the process is inspected to find the cause of this problem. Each of the songs collected from the billboard data is only organised by song name, the high ambiguousity of the song names can cause certain data distortion. An example is found to confirm this theory: The billboard hit "Get Together" is created by The Beatles, however John Cipllinos and Nick Gravenites have also collaborated on a song with the same name. This song is not on the billboard but has the exact same name as the one created by The Beatles. The solution is to organise the billboard hits by song and the corresponding artist. However this causes the next problem.

The collected data is consciously filtered by single artists, which means any song on the billboard that is produced by a band will not be matching the data at all. There is a difference between the collaborating musicians work together in a band or if they choose to collaborate with another musician on their own solo career. This problem is considered in the data processing phase. The main problem is that the band data from Rovi database does not contain enough individual musician information to use in the network construction. On the other hand, the artist data from Rovi database does not contain the songs he/she created with the different bands. A possible solution of this is to further refine the data by attaching the band(s) as an extra parameter for each musician in the database. Then the program can be altered to check for band names whenever collaboration is found.

The construction of the collaboration network also contains certain inaccuracy. Rovi database only indicates a collaboration relation by a simple score and a name, without the product in terms of songs for the specific collaboration. This causes problems to check whether the collaboration network is valid for the time related network analysis. The time related data is produced by checking whether the collaborator of a certain artist is also active in the same decade. This could cause data distortion by the specific scenario that if two musicians are both active from the 1960s, and collaborate together in the 1990s, they would be added in the 1960s collaboration network. The most obvious solution is to match the songs of both artists and construct a song list to check which songs they have collaborated together and further use the API call to find the correct decade of the songs. However it is encountered by the problem that the song data of the musicians indicates a certain degree of incompleteness. An examples of this is that the most degree central musician in the collaboration network – the session drummer Jim Keltner with 48 different collaborators, only has 6 songs according to the Rovi database. This makes it impossible to apply the mentioned solution to the data based on the collected dataset. However it

is possible to further data-mine other websites to seek out the missing data and refine the dataset.

6.3 Future Work

6.3.1 Data Expansion

The expansion of database is on a high priority for further work on this project. All the previously mentioned data distortion can be diminished by the expansion. However the data would need to be processed with algorithms of higher complicity.

By collecting data of artists from other genre than Pop/Rock, it is also possible to expand the analysis on genre specific results as well as cross genre comparison.

6.3.2 Network Interpretation

It is possible to conduct more in depth analysis with the current dataset. Comparison between the degree distribution and the local clustering of the development analysis can indicate a difference of similarity based on the best fitting regression line. Both of these are plotted and can be viewed in the Appendix H. However the results of the comparison are yet to be completed to be used within the report. These results could provide valuable contribution to the development analysis and interpreting the difference between realworld and artificial networks.

It is also interesting to apply the link clustering algorithm on the networks to gain certain insights of the group partitions between the musicians, which could contribute to either the understanding of the networks or form further recommendations upon.[1] The link clustering algorithm is applied on the existing dataset but is yet to be process with precision to be included in the thesis. The files created by the algorithms can be found on the following link https://www.dropbox.com/s/yeeeodrcp12y834/Link%20Clustering.rar.

6.3.3 Prototype Development

First step is to improve the prototype layout problems reported in the user feedback process. The next problem is the Artist List page, which has a noticeable long loading time. It is reasonable to hide the list elements before the search bar is activated by the user. However this does not solve the loading time issue completely, since the listview items still needs to be processed. It is then possible to utilize a search bar that resembles to the autocomplete function on a database containing the musicians rather than use the filtered list to store all the names as list items. This approach would fetch the necessary data only when it is required. Therefor it will improve the performance of the search function.

The application size will also become a noticeable problem with data expansion. Every Artist Profile page Html code is stored locally on the application. It is possible to predict that when the database vastly increases in its size, the application would require too much harddisk space to be installed. This problem can be solved by uploading all Artist Profiles to an online database, so the program simply refers to this database with the search function and fetch data from this database to shown individual artist profile and recommendations.

To increase the interaction and graphic level of the application, it is viable to construct interactive network graphs, provided that they contain a certain level of simplicity. All the relation between musicians are stored as JSON format files to allow visualization by using the forced directed graph on D3js.org

The influence network is concluded to contain properties of a hierarchical tree, which can be implemented as an interactive visualization to enable the functionality of graphic musician exploration.

6.3.4 Transform The Prototype To Music Player

This would require altering the fundamental purposes of this prototype. It is shown in the user feedback that a recommendation function should contain more songs results including the full version of songs. To do so, the purpose of the prototype has to be changed to play music, while the network recommendation approach is implemented as a functionality of this music player. This means the application have to access to an online database with licenses to the full version of songs.

It requires a lot of resources and planning but this would increase market value significantly and contribute to promoting such a recommendation system.

Chapter 7

Conclussion

This thesis presented a new and functional approach to the existing music recommendation systems. The existing systems are based on either directly comparing songs to each other or utilize user models to study the pattern of behaviours. The approach described in this thesis is based on the social networks between the musicians. By investigating the networks, several insights are revealed and ultilized to form the recommendation content. In this approach, the user has direct control of navigating through the recommendation contents.

This approach is implemented as a prototype on both of the discussed smartphone platforms. The test results of the prototype validated the functionality and certain market value as well as indicating the significant potential for further development. The current prototype is able to perform the use cases, however the performance is limited by the current completeness of the prototype.

Further development can result in an application that provides a very different experience on exploring through music rather than a simple function within a music player. The users would gain knowledge of the musicians' background as well as discover new styles of music to listen to. The user feedbacks for the current prototype validate the existing market value as well as providing information and suggestions for improve the design and functionality of this prototype. New functionality can be developed with further study of the networks and new design can be created by using the general feedback from the

users.

All the initial project goals are achieved and the prototype is expected to be improved to the presentation.

Appendix A

Mock ups for Artist Profile

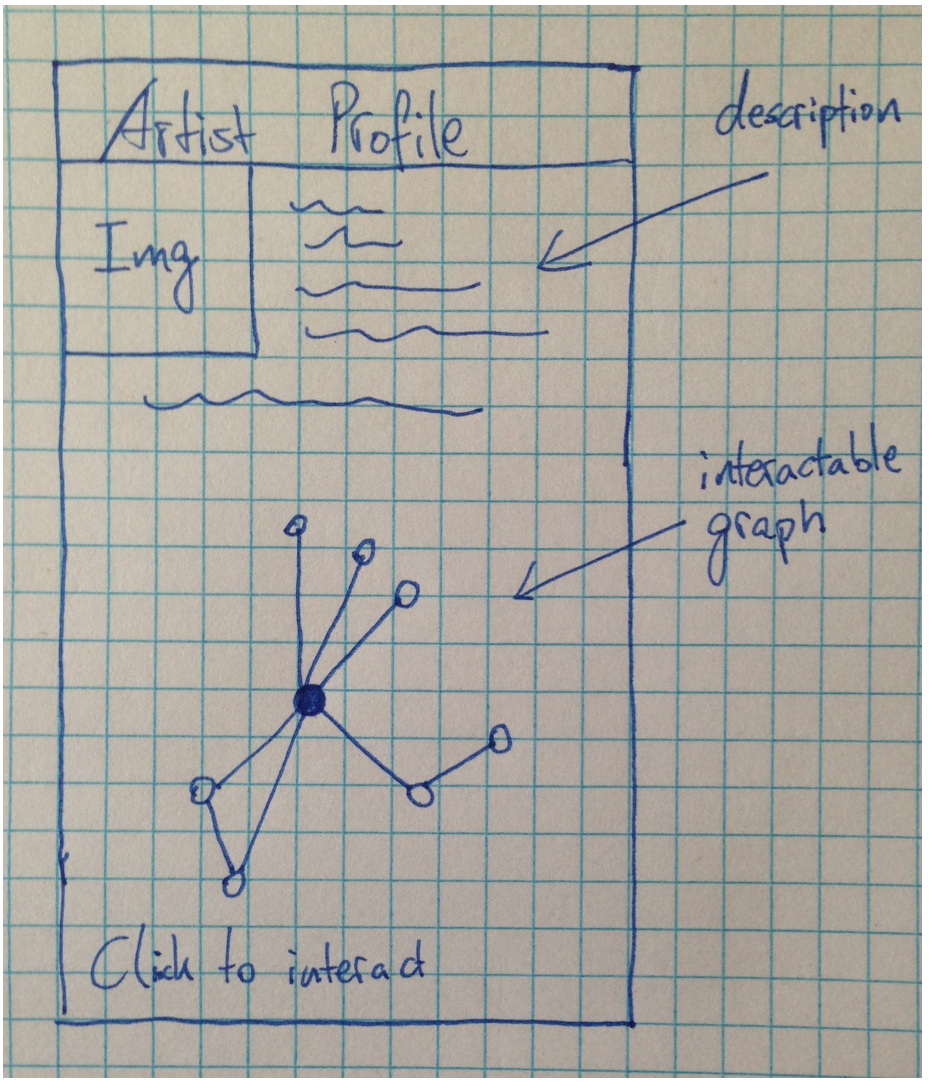


Figure A.1: This is the first mock-up for the Artist Profile page

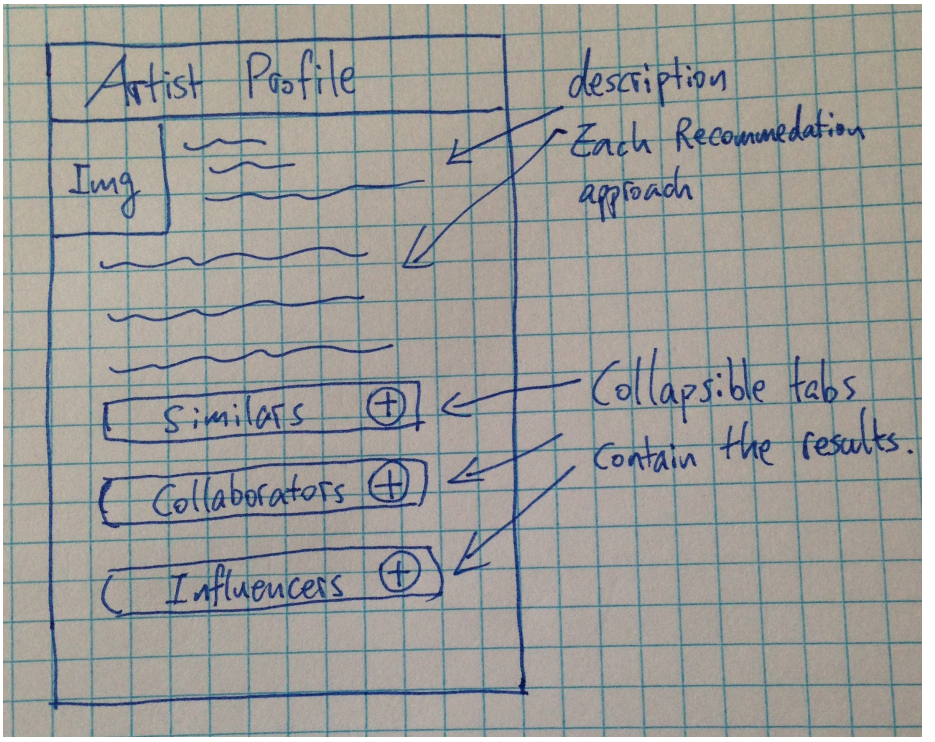


Figure A.2: This is the second mock-up for the Artist Profile page

Appendix B

Artist Data

Attribute	Format	Description	Attribute	Format	Description
name	String	The full name of the artist	**similar**	List of dict	List of all the musicians similar to this artist
headlineBio	String	A short description of this artist	**collaborator-With**	List of dict	List of all the musicians who collaborated with this artist
active	List	The list of active decades of this artist	**influencer**	List of dict	List of all the musicians who influenced this artist
isGroup	Boolean	True if the data belongs to a group, false if the data belongs to single artist	**songs**	List of dict	List of all the songs by this artist
discography	List of dict	List of all the albums by this artist	**memberOf**	List of dict	List of all the bands this artist worked for
musicGenres	List of dict	List of all the genres the artist is related to			

Figure B.1: The most relevant attributes for the collected artist data

Appendix C

All graph visualizations

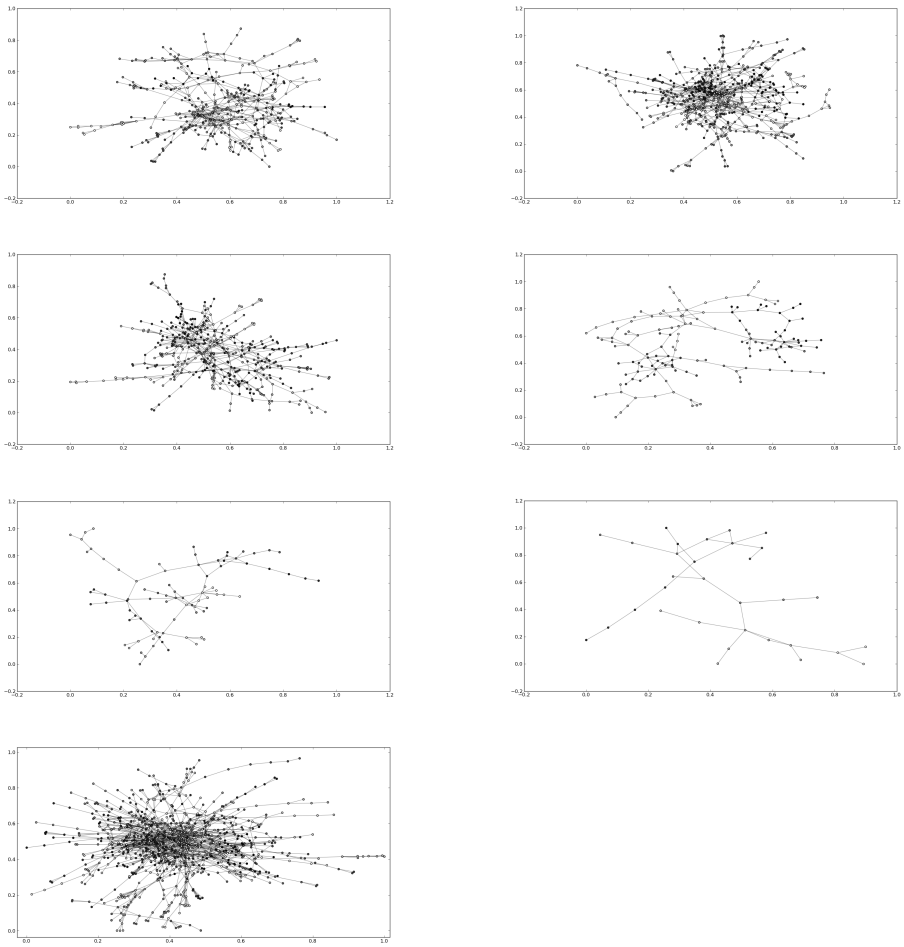


Figure C.1: The visualizations of collaboration networks in each decade. The top left graph is for 1960 and the bottom left graph is for the combined collaboration network

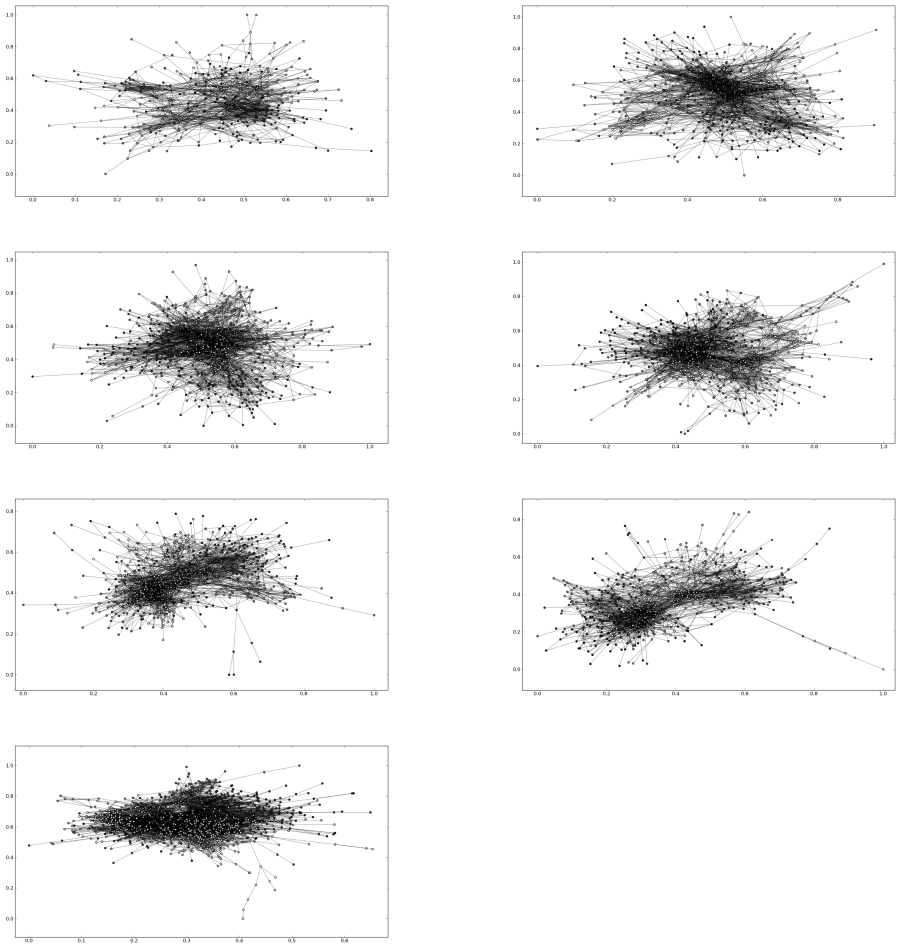


Figure C.2: The visualizations of similarity networks in each decade. The top left graph is for 1960 and the bottom left graph is for the combined similarity network

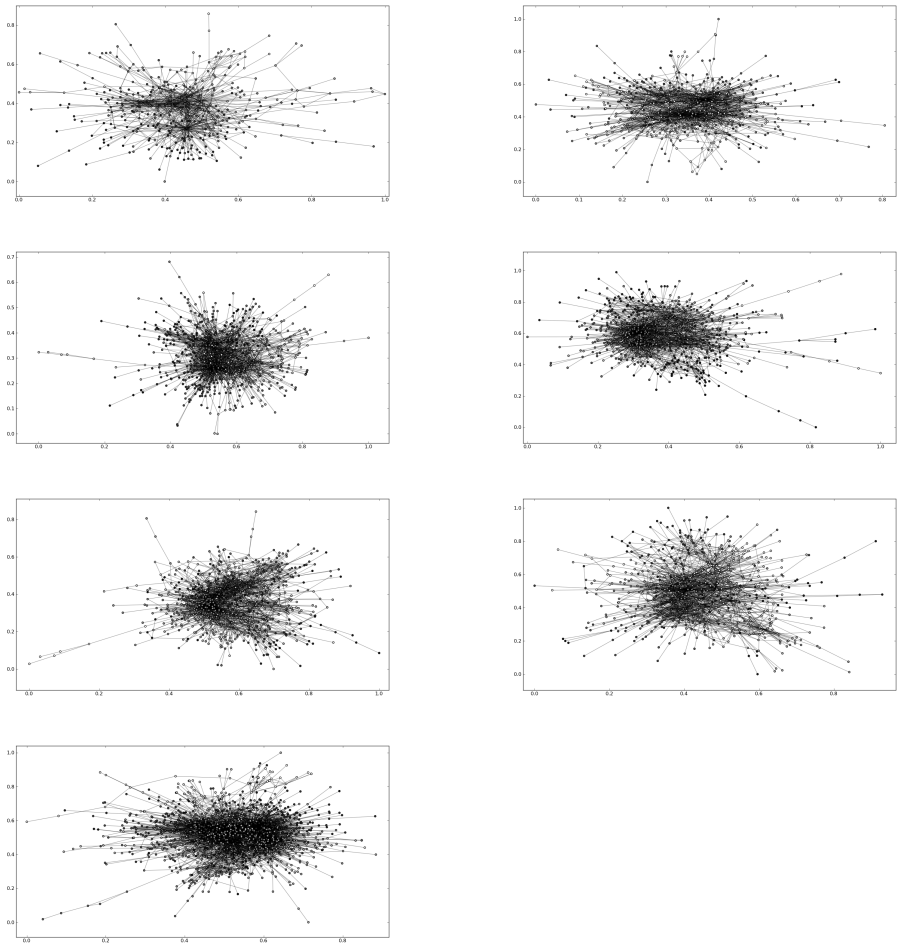


Figure C.3: The visualizations of influence networks in each decade. The top left graph is for 1960 and the bottom left graph is for the combined influence network

Appendix D

Development data

Collaboration	n	m	GC_n	GC_m	Com	MGP	C	R
1960	485(30,4%)	741	385(79,4%)	666(89,9%)	17	6,89	0,3	0,12
1970	701(39,9%)	1066	549(78,3%)	956(89,6%)	21	6,62	0,23	0,04
1980	687(36,7%))	862	410(60,1%)	644(93,7%)	19	7,1	0,25	0,007
1990	382(24,6%)	374	146(38,2%)	179(46,9%)	11	8	0,17	-0,004
2000	241(16,7%)	212	84(34,9%)	97(40,2%)	8	6,8	0,15	-0,06
2010	193(15,9%)	156	32(16,5%)	34(21,8%)	5	4,9	0,1	-0,07
All	1261(26,4%)	1951	907(71,9%)	1688(86,5%)	28	6,67	0,24	0,086

Influencer	n	m	GC_n	GC_m	Com	MGP	C	R
1960	416(21,2%)	1029	400(96,1%)	1021(99,2%)	12	3,5	0,07	-0,19
1970	601(34,2%)	1716	585(97,3%)	1708(99,5%)	8	3,4	0,07	-0,16
1980	757(40,3%)	2091	749(98,9%)	2087(99,8%)	13	3,7	0,06	-0,12
1990	754(40,3%)	2204	735(97,4%)	2194(99,5%)	13	3,6	0,07	-0,11
2000	765(48,6%)	2231	752(98,3%)	2224(99,7%)	11	3,6	0,07	-0,11
2010	685(53,1%)	1939	681(99,4%)	1937(99,9%)	9	3,6	0,07	-0,12
All	1436(30,1%)	4510	1395(97,1%)	4488(99,5%)	11	3,8	0,06	-0,11

Similarity	n	m	GC_n	GC_m	Com	MGP	C	R
1960	632(39,7%)	2011	587(92,9%)	1981(98,5%)	11	4,1	0,24	0,19
1970	931(53,0%)	3501	887(95,2%)	3470(99,1%)	14	4	0,24	0,18
1980	1105(58,9%)	4084	1063(96,2%)	4060(99,4%)	12	4,2	0,19	0,22
1990	1013(63,3%)	3904	994(98,1%)	3893(99,7%)	12	4,2	0,2	0,21
2000	1044(72,5%)	3983	1018(97,5%)	3968(99,7%)	16	4,3	0,2	0,2
2010	952(78,5%)	3464	934(98,1%)	3453(99,7%)	13	4,4	0,21	0,18
All	2290(48,1%)	8506	2172(94,8%)	8426(99.5%)	16	4,6	0,18	0,16

Figure D.1: This is the data computed for all the decade specified graphs. n is the number of nodes(procentage of the files in the decade). m is the number of edges. GCn and GCm are the nodes and edges of the giant component. Com is the number of communities. MGP is the mean geodesic path. C is the global clustering coefficient and R is the assortativity

Appendix E

Succesful Collaborations

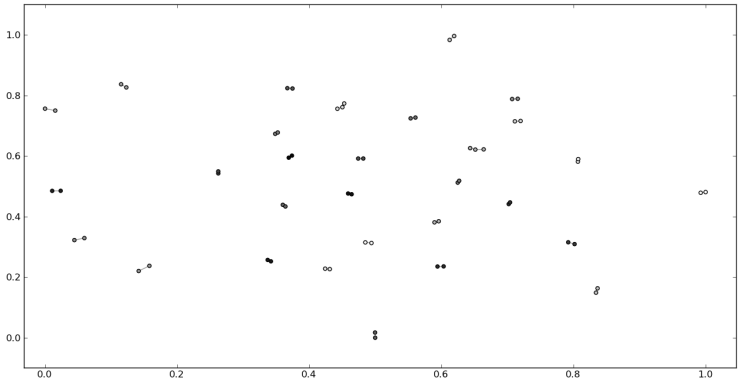


Figure E.1: This is the network constructed based on the musicians with matched song data

Collaborator1	Collaborator2	Number of billboard hits
Jon Anderson	Rick Wakeman	1 (21)
Dan Fogelberg	Tim Weisberg	2 (40)
Leon Russell	Willie Nelson	5 (20)
Buddy Miles	Carlos Santana	1 (6)
Colin Blunstone	Rod Argent	6 (26)
Marty Balin	Paul Kantner	4 (26)
Jah Wobble	Jaki Liebezeit	1 (4)
John Cipollina	Nick Gravenites	1 (19)
Andy Mackay	Phil Manzanera	1 (33)
Gordon Giltrap	Rick Wakeman	1 (13)
Chris Spedding	Robert Gordon	11 (53)
Richard Manuel	Rick Danko	3 (34)
Ronnie Lane	Steve Marriott	3 (38)
Burt Bacharach	Hal David	4 (20)
Dan Penn	Spooner Oldham	2 (14)
Atticus Ross	Trent Reznor	1 (58)
Bill Bruford	Tony Levin	1 (12)
Carla Olson	Gene Clark	3 (43)

Figure E.2: This is the output from cross-referencing the billboard

Appendix F

Recommendation output

```
...
Enter artist name: Eric Clapton
1 similar collaborators are found, displaying top 10
[('Pete Townshend', 4)]
-----------------------------------------------------------------------------------------

3 influencer with similarity to this artist found, displaying top 10
[('J.J. Cale', 16), ('B.B. King', 14), ('Bob Dylan', 14)]
-----------------------------------------------------------------------------------------

7 follower with similarity to this artist found, displaying top 10
[('Joe Bonamassa', 17), ('Jeff Healey', 16), ('Peter Frampton', 16), ('Joe Walsh', 15),
('Mark Knopfler', 14), ('Chris Rea', 14), ('Stevie Ray Vaughan', 14)]
-----------------------------------------------------------------------------------------

1 follower who is not similar to this artist is found, displaying top 10
[('Eddie Van Halen', 9)]
-----------------------------------------------------------------------------------------

13 artists have common influencer as the input artist, and is also similar to the input
artist, displaying top 10
[('Stevie Ray Vaughan', 18), ('Bonnie Raitt', 18), ('Duane Allman', 18), ('Jimi Hendrix
', 18), ('Johnny Winter', 18), ('Jeff Beck', 18), ('Keith Richards', 16), ('Robbie Robe
rtson', 16), ('John Lennon', 16), ('Neil Young', 15)]
-----------------------------------------------------------------------------------------

79 artists have common influencer as the input artists, but are not similar to the inpu
t artist, random 10 is displayed
[('Nick Lowe', 9), ('Dave Alvin', 9), ('Cub Koda', 9), ('Jim Morrison', 9), ('Rick Dank
o', 9), ('Brian Setzer', 9), ('Del Shannon', 9), ('Van Morrison', 9), ('Doug Sahm', 9),
('Warren Zevon', 9)]
-----------------------------------------------------------------------------------------
```

Figure F.1: This is the output generated based on the musician Eric Clapton

Appendix G

User feedback

G.1 Questions

This is the question ark the users received.

G.1.1 Before testing the prototype

Please answer the following questions briefly before testing the prototype

1. Are you familiar with the music recommendation function on smartphone apps?
2. Do you have an understanding of how these recommendations are formed? If not, are you interested in the recommendation approach and understand why the songs are recommended?
3. Do you prefer songs that are similar to what you usually listen to or songs with a different style or genre?
4. If the songs should be different, would you prefer the recommendation to create random results or be based on a certain criteria?

5. Do you have certain musicians that you particularly like?
6. Would you be interested in music by your favourite musicians' collaborator, similar or influencers?

G.1.2 Testing the prototype

Unpack the Prototype.zip file to any location you want to and open the MusicGrapher.html to test the prototype. The current prototype only has data for Eric Clapton and the result from network analysis contain only text based information.

Please perform the following tasks, both tasks should be performed from the Home page:

- Find the musician: Eric Clapton, and acquire one artist from the recommendation Go back to the home page
- Find who has inspired most musicians

Please answer the following questions after the tasks are accomplished:

1. It is simple to perform the tasks?
2. Is it manageable to find the artist you're looking for?
3. Do you understand why the different artists are recommended to you?
4. Are the recommendations sufficient? What would you suggest to improve it?
5. Are the network results interesting or helpful for your experience?
6. Did you have any problems navigating in the application?
7. Do you think the artist profile information is sufficient? Any suggestions?
8. Do you like the design of this app?
9. Did you notice any problems functionality wise or design wise?
10. Other issues or thoughts?

G.2 Feedback

This is the combined feedback from the users. The individual feedback files can be viewed on https://www.dropbox.com/s/enoi5tsbhgrexau/User%20Feedback.zip.

G.2.1 Before test questons

Are you familiar with the music recommendation function on smartphone apps?

1. No
2. Some what
3. Yes, but rarely use them
4. No
5. Yes
6. Yes but not within actual apps

Do you have an understanding of how these recommendations are formed? If not, are you interested in the recommendation approach and understand why the songs are recommended?

1. No, based on similar content, but not interested as long as it works
2. Yes
3. No, but would like to know the approach
4. Vague insight, would like to understand them better
5. No, but woud like to know
6. Something with graphs, definetly interested

Do you prefer songs that are similar to what you usually listen to or songs with a different style or genre?

1. A mix of both - Mainly similar

2. Both
3. Depends on the mood, more comfortable with similar content but like to explore new style/genre occassionally
4. Similar
5. Both
6. Mostly similar, occasionally enjoy to explore

If the songs should be different, would you prefer the recommendation to create random results or be based on a certain criteria?

1. Based on criteria
2. Both
3. Both, mostly based on criteria
4. Certain criteria
5. Based on criteria
6. Based on criteria

Do you have certain musicians that you particularly like?

1. Yes
2. Yes
3. Yes
4. Yes
5. Yes
6. Yes

Would you be interested in music by your favourite musicians' collaborator, similar or influencers?

1. Yes

2. Yes
3. Probably
4. Yes, but depends on the relation detail
5. yes, would be a good way to explore music
6. Yes, very interested

G.2.2 Test questions

It is simple to perform the tasks?

1. Task1 - easy, task2 - difficult. No search function in results
2. Task1 - easy, task2 - difficult. Result page messy
3. Task1 - easy, task2 - slightly more advanced
4. Task1 - easy, task2 - would require more clear headings to find information quicker
5. Both tasks are easy
6. Task1 - easy, taske2 - slightly more advanced, very messy result page

Is it manageable to find the artist you're looking for?

1. Yes
2. Yes, the loading time on the artist list is a little long
3. Yes
4. Yes, search bar could be placed on the front page
5. Yes
6. Yes, especially because of the limited artist datbase

Do you understand why the different artists are recommended to you?

1. Yes

2. Yes
3. Yes
4. Yes, good and clear headings on the tabs
5. Yes
6. Yes

Are the recommendations sufficient? What would you suggest to improve it?

1. More songs from the suggested artists
2. Yes
3. Yes but would like more results in the same style
4. Yes, Could include a short description on the artist before navigation into his profile
5. Yes, but the recommendation tabs could be labeled simpler
6. Yes, could include active period for the recommendation

Are the network results interesting or helpful for your experience?

1. Not really
2. Yes
3. Yes, require more results
4. Yes
5. Yes, like the audio preview
6. Yes, could include more results and allow to explore in a visualized network

Did you have any problems navigating in the application?

1. No except the slightly unclear result presentation
2. No
3. No

4. No, back button on the artist list is too far down
5. No
6. No

Do you think the artist profile information is sufficient? Any suggestions?

1. Add country, birth/death
2. Add which band they have been a member of
3. It's okay
4. Good, could include top songs and albums
5. Yes, could include carrier summary and top songs
6. Yes, could include artist discography

Do you like the design of this app?

1. Yes, nice and simple
2. Yes
3. Yes, nice and clean
4. Yes, simple and smooth
5. Yes, the tabs could use a different color to highlight the functions
6. Yes, but too much white space on homepage

Did you notice any problems functionality wise or design wise?

1. Spelling mistakes
2. Delay on the artist search page
3. No
4. Loading the artist list is slow
5. Loading the artist list is slow

6. No

Other issues or thoughts?

1. Likes the headline bio
2. Would be good to be able to save favourites or view history of navigation
3. -
4. -
5. -
6. Would like to see a visualization of the network

Appendix H

Degree distribution and local clustering

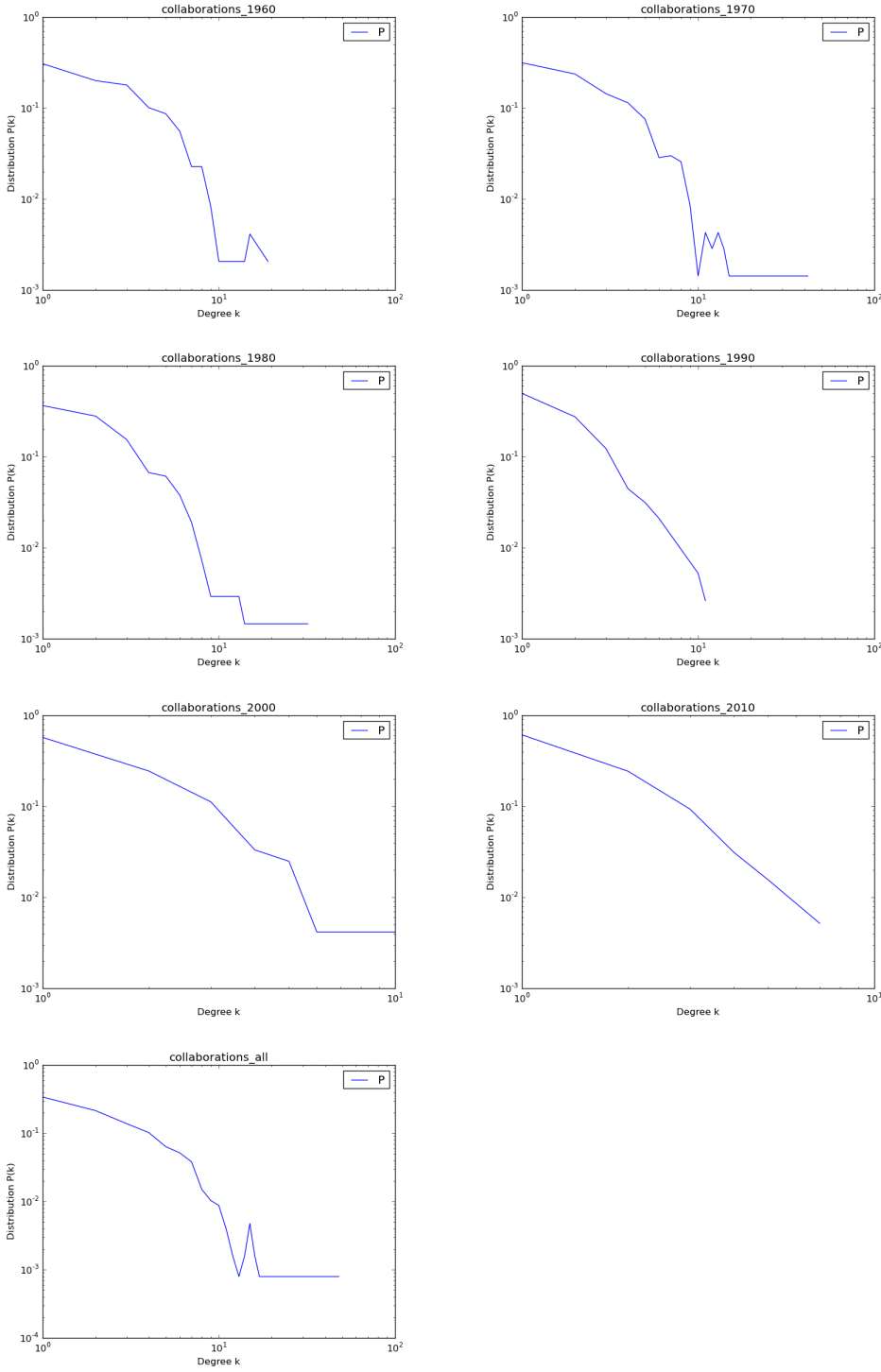


Figure H.1: The degree distribution of all the collaboration networks for every decade, starting from the top left is the graph for the 1960s and the bottom left is for the combined collaboration network

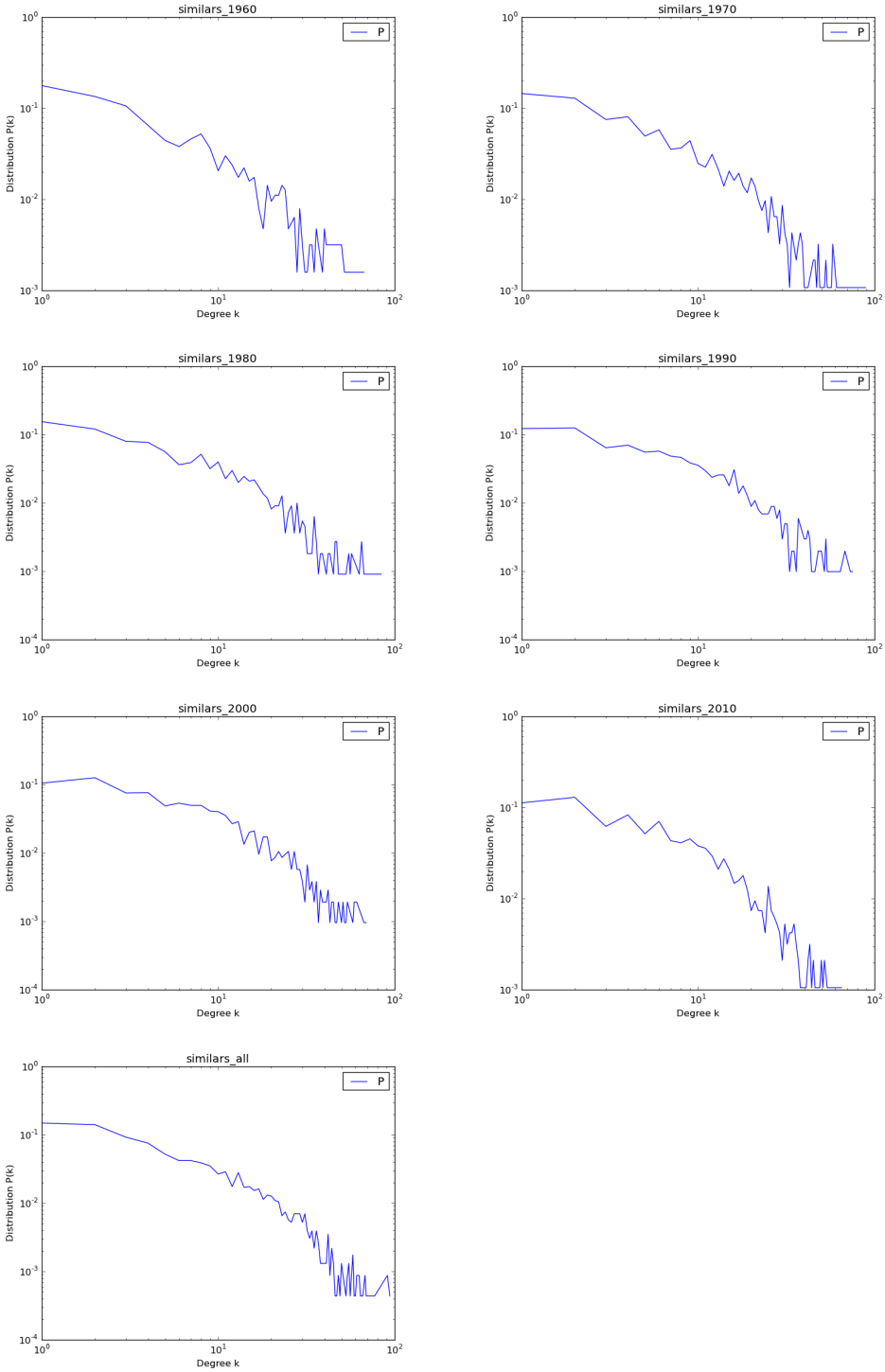


Figure H.2: The degree distribution of all the similarity networks for every decade, starting from the top left is the graph for the 1960s and the bottom left is for the combined similarity network

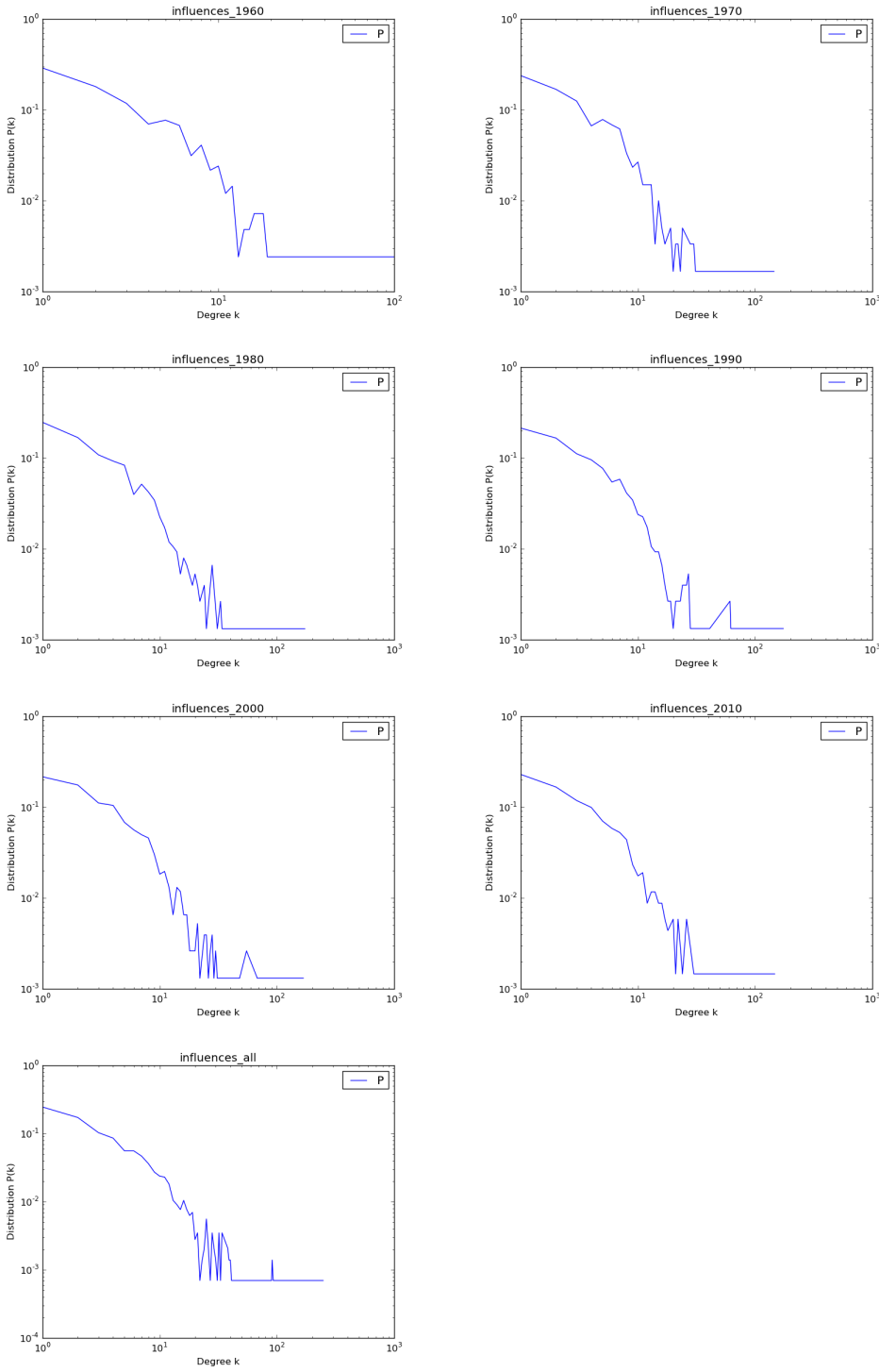


Figure H.3: The degree distribution of all the influence networks for every decade, starting from the top left is the graph for the 1960s and the bottom left is for the combined influence network

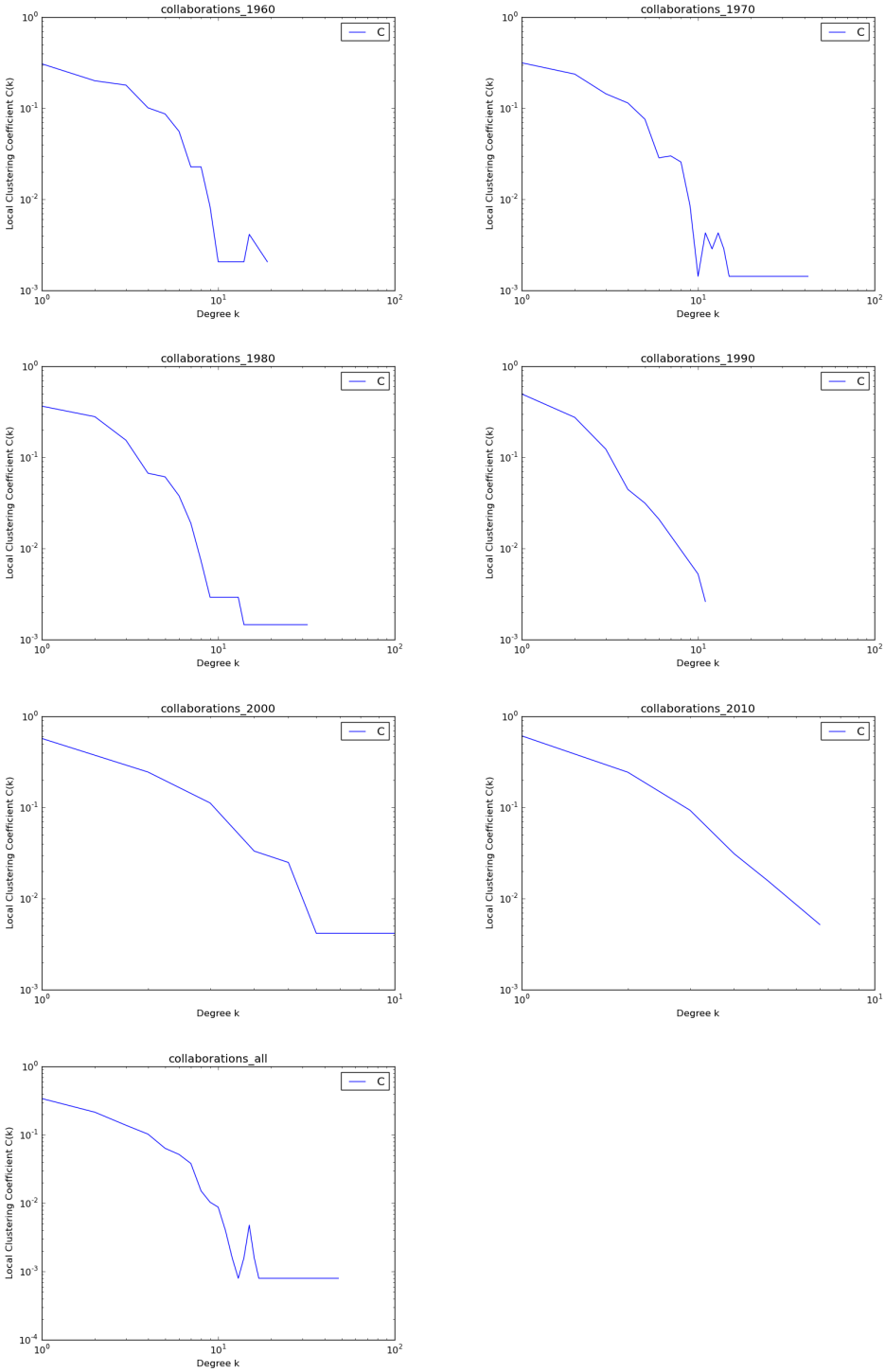


Figure H.4: The local clustering of all the collaboration networks for every decade, starting from the top left is the graph for the 1960s and the bottom left is for the combined collaboration network

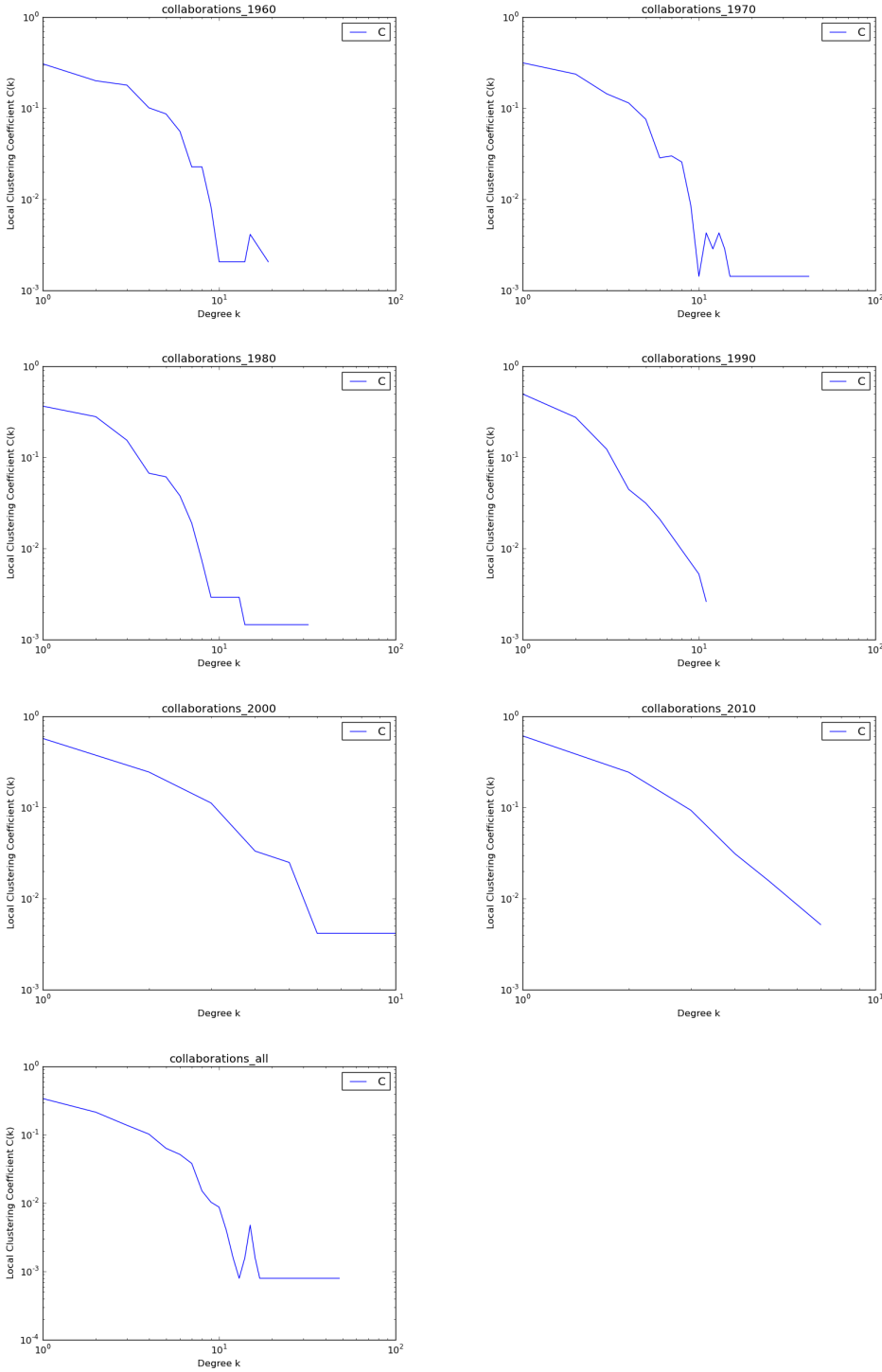


Figure H.5: The local clustering of all the collaboration networks for every decade, starting from the top left is the graph for the 1960s and the bottom left is for the combined collaboration network

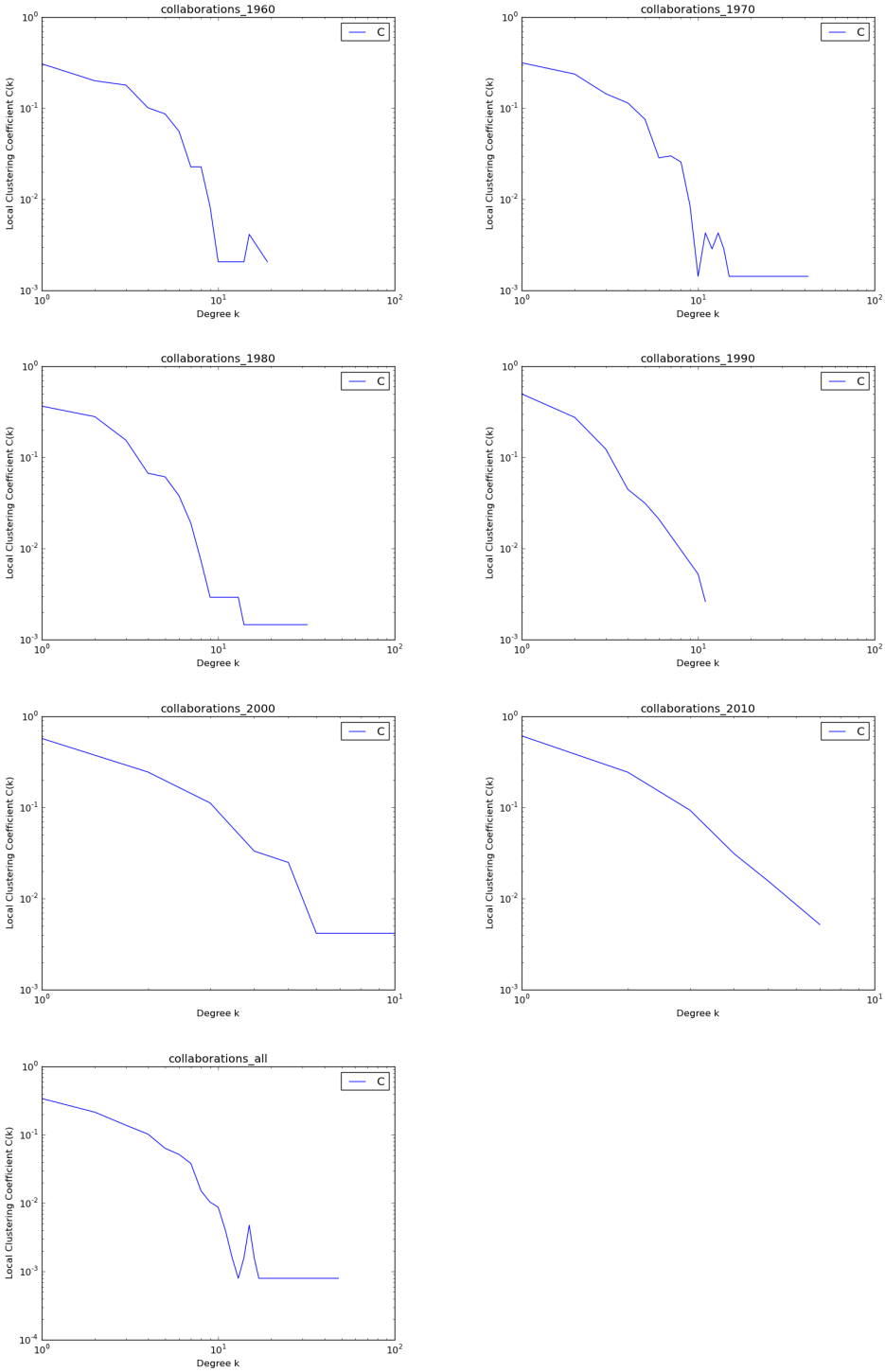


Figure H.6: The local clustering of all the collaboration networks for every decade, starting from the top left is the graph for the 1960s and the bottom left is for the combined collaboration network

Bibliography

[1] Link Clusering Algorithm. http://barabasilab.neu.edu/projects/linkcommunities/.

[2] Jon Kleinberg David Easley. Networks, crowds, and markets: Reasoning about a highly connected world [viewed 03-03-2013]. http://www.cs.cornell.edu/home/kleinber/networks-book/networks-book.pdf, 2010.

[3] EchoNest. http://echonest.com/.

[4] PhoneGap Getting Started Guides. http://docs.phonegap.com/en/2.7.0/index.html.

[5] IDC.com. Idc press release 14 feb 2013 [viewed 05-05-2013]. http://www.idc.com/getdoc.jsp?containerId=prUS23946013, 2013.

[6] Yehuda Koren. The bellkor solution to the netflix grand prize [viewed 01-05-2013]. http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf, 2009.

[7] Yoav Shoham Marko Balabanovic. Content-based, collaborative recommendation [viewed 21-04-2013]. http://dl.acm.org.globalproxy.cvt.dk/citation.cfm?id=245124, 1997.

[8] JQuerty Mobile. http://jquerymobile.com/.

[9] NetworkX. http://networkx.github.io/.

[10] Pandora. Music genome project@ [viewed 20-04-2013]. www.pandora.com/about/mgp.

[11] PhoneGap. http://phonegap.com/.

[12] Rovi. http://www.rovicorp.com/.

[13] Brian Whitman. How music recommendation works - and doesn't work [viewed 02-05-2012]. http://notes.variogr.am/post/37675885491/how-music-recommendation-works-and-doesnt-work, 2013.

[14] Wikipedia. Assortativity [viewed 07-02-2013]. http://en.wikipedia.org/wiki/Assortativity.

[15] Wikipedia. Clustering coefficient[viewed 07-02-2013]. http://en.wikipedia.org/wiki/Clustering_coefficient.

[16] Wimp. http://wimp.dk/wweb/index/.

[17] Trong Duc Le Anh Duc Duong Xuan Nhat Lam, Thuc Vu. Addressing cold-start problem in recommendation systems [viewed 01-05-2013]. http://dl.acm.org.globalproxy.cvt.dk/citation.cfm?id=1352837, 2008.