Goldsmiths UNIVERSITY OF LONDON

M. Phil to PhD Upgrade

Contextualized Music Recommendation:
The Playlist as Recommendation Engine

Supervisor:
Dr. Christophe Rhodes

Co-Supervisor:
Dr. Michael Casey

Author:
Benjamin Fields
b.fields@gold.ac.uk

Co-Supervisor:
Dr. Mark d'Inverno

Assessor:
Dr. Robert Zimmer

December 11, 2008

Abstract

A framework is described to consider various real world playlist use cases. Automatic playlist generation is introduced as a means to improve music recommendation. Literature in related topics is discussed.

A sample of the Myspace artist network is examined to investigate the relationship between social connectivity and audio-based similarity. Audio data from the Myspace artist pages is analyzed using well-established signal-based music information retrieval techniques. In addition to showing that the Myspace artist network exhibits many of the properties common to social networks, it is seen that there is an ambiguous relationship between audio-based similarity and the social connectivity. Further the Myspace sample is examined with the pairwise relational connectivity measure Minimum cut/Maximum flow. These values are then compared to a pairwise acoustic Earth Mover's Distance measure and the relationship is discussed. A means of constructing playlists using the maximum flow value to exploit both the social and acoustic distances is realized.

Two playlist generation methods are proposed for development and experimentation. The first is a direct extension of the myspace dataset analysis into a robust playlist system for interactive internet radio broadcast. The second is content based system which uses expert constructed playlists to construct transition models which can then be used on new material. This is followed by a discussion of evaluation needs and strategies.

Contents

List of Figures

List of Tables

List of Publications

B. Fields and M. Casey. Using audio classifiers as a mechanism for content based song similarity. In *Proc. Audio Engineering Society 123rd Int. Conv.*, New York, NY, USA, October 2007.

M. Mauch, S. Dixon, C. Harte, M. Casey, and B. Fields. Discovering chord idioms through beatles and real book songs. In *Int. Symposium on Music Information Retrieval*, Vienna, Austria, September 2007.

B. Fields, K. Jacobson, M. Casey, and M. Sandler. Do you sound like your friends? exploring artist similarity via artist social network relationships and audio signal processing. In *Int. Computer Music Conference*, August 2008.

B. Fields, K. Jacobson, C. Rhodes, and M. Casey. Social playlists and bottleneck measurements : Exploiting musician social graphs using content-based dissimilarity and pairwise maximum flow values. In *Proc. of Int. Symposium on Music Information Retrieval*, September 2008.

K. Jacobson, B. Fields, and M. Sandler. Using audio analysis and network structure to identify communities in on-line social networks of artists. In *Proc. of Int. Symposium on Music Information Retrieval*, 2008.

Chapter 1

Introduction

As freely–available audio content continues to become more accessible, listeners require more sophisticated tools to aid them in the discovery and organization of new music that they will find enjoyable. This need, along with the advent of internet based social networks and the steady progress of signal based Music Information Retrieval have created an opportunity to exploit both social relationships and acoustic similarity in recommender systems. These domains can provide the means to contextualize recommendation through the generation of playlists.

Motivated by this, the Myspace artist network is examined. Though there are a number of music oriented social networking websites, Myspace[1] has become the *de facto* standard for web-based music artist promotion. Although exact figures are not made public, recent estimates suggest there are well over 7 million artist pages[2] on Myspace. For the purpose of this paper, *artist* and *artist page* are used interchangeably to refer to the collection of media and social relationships found at a specific Myspace page residing in Myspace's artist subnetwork, where this subnetwork is defined as those Myspace user pages containing the audio player application.

The Myspace social network, like most social networks, is based upon relational links between *friends* designating some kind of association. Within each Myspace user's friends there is a subset of between 8 and 40 *top friends*. While all friends are mutually confirmed, individual users unilaterally select top friends. Additionally, pages by *artists* will usually contain streaming and downloadable media of some kind either audio, video or both.

Social networks of this sort present a way for nearly anyone to distribute their own media and as a direct result, there is an ever larger amount of available music from an ever increasing array of artists. Given this environment of content, how can we best use all of the available information to discover new music? Can both social metadata and content based comparisons be exploited to improve discovery of new material?

To work towards answers to these and related questions, the relationship between the connectivity of pairs of artists on the Myspace top friends artist network and a measure of acoustic dissimilarity of these artists is explored. This is followed by ways to improve upon the sample data used in the initial experiments. Two prototype systems of music playlist generation are then presented with attention paid to means for evaluation.

[1] `http://myspace.com/`

[2] `http://scottelkin.com/archive/2007/05/11/MySpace-Statistics.aspx` reports as of April 2007 ∼25 million songs, our estimates approximate 3.5 songs/artist, giving ∼7 million artists

Immediate following this chapter is an overview of objectives to be achieved by the conclusion of this body of research. Chapter 2 discusses a brief summary of real world use cases of playlists along with a review of relevant literature from complex network theory, signal-based and textual music analysis. The initial experiments into the relationship between the social connectivity and the acoustic feature space and their results are presented and discussed in Chapter 3. The implications and direction for future work are discussed in Chapter 4. Finally a research plan is presented describing a timeline to achieve these objectives.

Objectives

1. A complete playlist generation system which takes as input a variable number of songs and/or stylistic designations and produces as output an ordered list of songs and upon request, a *mixed* continuous audio stream of these songs.

2. An evaluation system to assess the success of the above system's output for at minimum a subclass of results produced for well-formed and specific playlist use cases.

3. Extending existing methods of determining relationships (e.g. similarity, sequentiality) between pairs of songs in order to better use these relationships to create playlists in the system created to meet objective 1.

Chapter 2

Background

The use of context in the form of a playlist as a music recommendation is the driving backdrop behind this research. This will be explored in Section 2.1 along with a framework for considering various forms of real-world playlist use cases.

Following, a brief review of complex network theory, traditional graph metrics, network flow analysis and signal-based music analysis in Sections 2.2 and 2.3. These disciplines apply intuitively to Music Information Retrieval; however, only recently the two have been applied simultaneously to the same data set. There are however relevant works investigating to some measure the interaction between disparate feature spaces in music. These works are reviewed in Section 2.4

2.1 The Playlist As Recommender

In many ways the process of putting together a playlist given a specific set of requirements can be considered as a specific case for a music recommender system. This allows a much wider set of existing literature can be exploited. While the idea of viewing a recommendation in context is not as of yet common place in the domain of music, it has been exploited in recommender systems [Anand and Mobasher, 2007] and in text information retrieval [Dunlop and Crossan, 2000]. Most commonly, some predictive text systems use the entire preceding sentence to make more informed guesses about the word being typed [Cohen and Singer, 1999, Stocky et al., 2004], allowing more efficient use of larger dictionaries. In many respects the playlist can be viewed as an analog to a sentence of text. By observing and exploiting what has come previously (or what is known to be coming next) a recommendation system can eliminate many otherwise possible candidates for recommendation. The other related side of context based recommendation is what can be considered *boosting*. In boosting, a recommendation that is acceptable independent of any context becomes much stronger when presented in context with supporting data points. In the music domain, this is consciously done by club DJs and radio show presenters [Bewster and Broughton, 2006]. It may also be present in other forms of human playlist generation, though this is less documented.

2.1.1 Relational Playlist Types

A playlist defines a wide variety of forms of music presentation. They can encompass nearly any ordered lists of songs. In order to discuss how playlists function in different contexts and situations, it is beneficial to categorize playlists based on common traits. What follows divides various real world forms of playlists sorted by the relationship between the entity creating the playlist(e.g. the playlist's *producer*) and the intended entity to listen to the playlist (e.g. the playlist's *consumer*). Examples of each category will be discussed along with common characteristics of each class. These categories and their common members can be seen in Table 2.1.

Expert to Listener	**Peer to Peer**	**Listener to Self**
Club DJ	Mixtape	Digital Library Playlists
Radio DJ	Web Published Playlists	Portable Player Playlists
Commercial mix CD		

Table 2.1: Various kinds of playlists classified based on the relationship between producer and consumer.

Expert to Listener: The Curator

This category encompasses any and all cases where the generation of the playlist occurs by a professional with the intended playback to occur via some medium which will allow many people to listen in parallel. Common examples that fit in this category include various forms of performance by disc jockey (DJ), (*i.e.* Radio Specialty Show, Standard Rotation Music Radio Show, Club DJ, etc.). In each of these use cases the basic idea is the same. From some finite set of songs a subset is selected and ordered based on the particular requirements and goals of the use case. The key factors that tends to subdivide this category are the total size of the collection from which songs are drawn and particular kind of feedback the expert receives from their audience.

There is another form within this category which has emerged with the widespread adoption of personal digital music player (*i.e.* iPods), the *book of playlists* [Ellingham, 2007, Lynskey, 2008]. These are playlists compiled by experts with content similar to a greatest hits CD or a compilation CD based on a topical idea. Unlike these compilation albums, the experts do not provide the tracks in these playlists. That is an exercise left to the reader.

Peer to Peer: A Contextualized Recommendation Among Friends

The Peer to Peer playlist is inherently social. These use cases encompass devices for peers to recommend not simply music but a context into which the recommended music is best understood or appreciated. Classically this is encapsulated in the mixtape, a cassette tape of tracks recorded off various records creating a personalized compilation. In the internet age, the same idea is realized over the internet, where any number of website and internet aware applications allow for the broadcasting and sharing of playlists (*i.e.* `http://mystrands.com`, `http://webjay.com`, iTunes, etc.)

Listener to Self: Everyone is a DJ

The simplest relationship between producer and consumer of playlist is when they are the same entity. This is seen most readily in a user of digital media management software such as iTunes or Windows Media Player creating playlists within the software for any number of purposes. The listener to self use case is less about discovery of new media and more concerned with exploiting know content to its fullest, since the songs which create the collection are all the personal collection of the listener.

Functional playlists

Functional playlists encapsulate selection and order of music for playback in non-listening environments. In other words, a functional playlist is any playlist that serves as *background* while some other function is taking place. This can best be seen commercially in the work of *i.e.* Muzak[1]. While this is perhaps the most common way playlists are used in the modern world [Lanza, 2004], it also must be considered outside the scope of this document. This is due to functional playlists being the furthest a *playlist* gets from a *recommender system* as the goal is for the listener to do whatever other task they were doing anyway, with the music only subconsciously noticed [Lanza, 2004].

2.1.2 Automatic Playlist Generation

There is a small body of work concerning the automatic generation of playlists. Most of these previous attempts at playlist generation rely on either textual metadata or content based features. There are limited current works attempting to merge these two data sources, which will be discussed in Section 2.4. Further, the principal interface for specifying a playlist through most previous works is by example, via a single song which starts the playlist.

Textual Metadata Generation Techniques

In work fairly indicative of textual metadata playlist generation [Platt et al., 2002] uses gaussian process regression, a form of meta-training, to formulate a kernel based on a text metadata hand entered describing a large corpus of albums. This kernel is then used to generate Listener to Self type playlists which are shown to better predict a listener's playlists then a kernel which has been designed by hand. While this method is only lightly evaluated it shows promise. Like any system based purely on textual metadata, this system can only be as good as the metadata of the user's collection. It is easy to see that a system such as this would not do well with a set of poorly formed, missing or incorrect metadata.

Another variant of the textual metadata case uses social relationships and common taste to make tailored playlists through collaborative filtering [Avesani et al., 2002, Hayes and Cunningham, 2004]. In this approach, by finding users with overlapping interests, recommendations can be made based on the strength of feedback from others with similar interests who have already listened to unknown works.

[1] `http://www.muzak.com`

Content Based Generation Techniques

There have also been a few notable content based playlist generation techniques. The basic premise in the simplest case of is to take an acoustic distance measure between pairs of songs in a collection and when given a seed song return an ordered list of its nearest neighbors as an ordered playlist (*i.e.*[Aucoutier and Pachet, 2002, Ragno et al., 2005])In Pampalk et al. [2005] a playlist generation method is described which relies on an acoustic distance to determine a song's nearest neighbors. This nearest neighbor list is presented as the user's initial playlist, which is then steerable via the use of a skip button. The skip button allows the recomendations to get better over time. These playlists are roughly analogous to a playlist somewhere in between a Peer to Peer playlist and the Expert to Listener case. Flexer et al. [2008] presents another interesting novelty to a purely content based approach. This work uses both a start and an end song as input. The algorithm then constructs a path through the collection which aims to minimize the average acoustic distance between each step of the playlist. The result is then evaluated against an objective genre measure and a subjective listener test via radio broadcast.

2.1.3 Playlist Specification: Query Interface

One common feature among many of the playlist generation methods discussed in Sections 2.1.2 and 2.1.2 is the means by which you specify a query. Most automated playlist generation methods ask for a single song to be used as the seed and use no other input from the user to determine the output playlist. It should be apparent that if a playlist is an ordered set, some means of specifying the order is needed as input, e.g. a playlist requires not only a position to start, but also a trajectory. Pampalk et al. [2005] and Flexer et al. [2008] are notable exceptions as they have other input mechanisms to allow a user to specify a starting position and a trajectory of some kind.

2.2 Tools to Analyze and Exploit Networks

2.2.1 Complex Networks

Complex network theory deals with the structure of relationships in complex systems. Using the tools of graph theory and statistical mechanics, physicists have developed models and metrics for describing a diverse set of real-world networks – including social networks, academic citation networks, biological protein networks, and the World-Wide Web. In contrast to *simple networks*, all these networks exhibit several unifying characteristics such as small worldness, scale-free degree distributions, and community structure [Newman, 2003]. Some definitions and concepts are briefly introduced below that will be used in this work.

A given network G is described by a set of *nodes* N connected by a set of *edges* E. Each edge is defined by the pair of nodes (i, j) it connects. This pair of nodes are *neighbors* via edge $E(i, j)$. If the edges imply directionality, $(i, j) \neq (j, i)$, the network is a *directed network*. Otherwise, it is an *undirected network*. In this paper, all edges are directed unless otherwise stated. In some graphs each edge $E(i, j)$ will have an associated label $w(i, j)$ called the *weight*. This weight is sometimes thought of as the cost of traversing an edge, or an edge's resistance. The number of edges incident to a node i is the *degree* k_i. In a directed network there will be an *indegree* k_i^{in} and an *outdegree* k_i^{out} corresponding to the number of edges pointing into the node and away from the node respectively.

The *degree distribution* $P(k)$ of a graph is the proportion of nodes that have a degree k. The shape of the degree distribution is an important metric for classifying a network – *scale-free* networks have a power-law distribution $P(k) \propto k^{-\gamma}$, while *random* networks have a Poisson distribution [Newman, 2003]. Many real-world networks are approximately scale-free over a wide range of scales. Conceptually, a scale-free distribution indicates the presence of a few very-popular *hubs* that tend to attract more links as the network evolves [Costa et al., 2007, Newman, 2003].

Two nodes i and j are connected if a path exists between them following the edges in the network. The path from i to j may not be unique. The *geodesic path* d_{ij} is the shortest path distance from i to j in number of edges traversed. For the entire network, the *average shortest path* or mean geodesic distance is l.

$$l = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i \geq j} d_{ij} \tag{2.1}$$

where d_{ij} is the geodesic distance from node i to node j and n is the total number of nodes in the network. In a *small-world* network the mean geodesic distance is small relative to the number of nodes in the network [Costa et al., 2007, Newman, 2003]. The largest geodesic distance in a network is known as the *diameter*.

2.2.2 Network Flow Analysis

The basic premise in network flow analysis is to examine a network's nodes as sources and sinks of some kind of *traffic* [Ahuja et al., 1993]. Typically, though not exclusively, flow networks are directed, weighted graphs. A simple flow network can be seen in Figure 2.1. Many useful strategies for determining the density of edge connectivity between sources and sinks can be found in this space[Nagamochi and Ibaraki, 1992]. One of the most common among them is the Maximum Flow/Minimum Cut Theorem[Elias et al., Dec 1956], which is a means of measuring the maximum capacity for fluid to flow between a source node to a sink node or, equivalently, the smallest sum of edge weights that must be *cut* from the network to create exactly two subgraphs, one containing the source node and one containing the sink node. This will be discussed in more detail in Section 3.3. The few examples of network flow type analysis in music deal primarily with constructing playlists using partial solutions to the Traveling Salesman Problem [Knees et al., 2006] or use exhaustive and explicit metadata[Alghoniemy and Tewfik, 2001].

2.2.3 Musician Networks

Quite naturally, networks of musicians have been studied in the context of complex network theory – typically viewing the artists as nodes in the network and using either collaboration, influence, or similarity to define network edges. These networks of musicians exhibit many of the properties expected in social networks [Cano et al., 2006, Gleiser and Danon, 2003, Park et al., 2007]. However, these studies all examine networks created

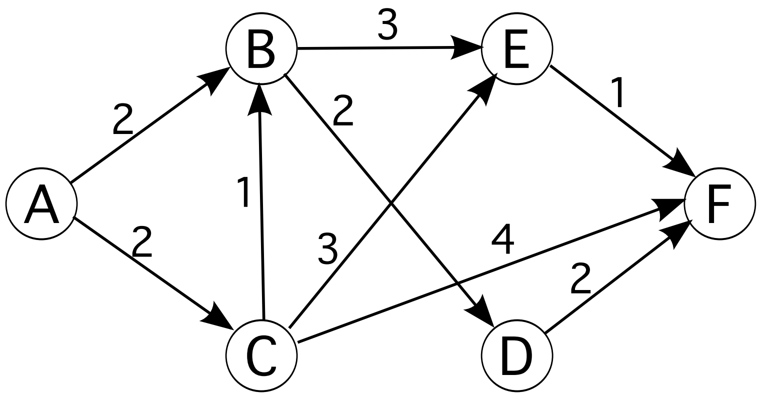


Figure 2.1: A simple flow network with directed weighted edges. Here the source is node A and the sink is node F.

by experts (*e.g.* All Music Guide[2]) or via algorithmic means (*e.g.* Last.fm[3]) as opposed to the artists themselves, as is seen in Myspace and other similar networks. Networks of music listeners and bipartite networks of listeners and artists have also been studied [Anglade et al., 2007, Lambiotte and Ausloos, 2006].

2.3 Content-Based Music Analysis

Many methods have been explored for content-based music analysis, attempting to characterizing a music signal by its timbre, harmony, rhythm, or structure. One of the most widely used methods is the application of Mel-frequency cepstral coefficients (MFCC) to the modeling of timbre [Logan, 2000]. In combination with various statistical techniques, MFCCs have been successfully applied to music similarity and genre classification tasks [Berenzweig et al., 2004, Logan and Salomon, 22-25 Aug. 2001, Pampalk, 2006].

A common approach for computing timbre-based similarity between two songs or collections of songs creates Gaussian Mixture Models (GMM) describing the MFCCs and comparing the GMMs using a statistical distance measure. Often the Earth Mover's Distance (EMD), a technique first used in computer vision [Rubner et al., 2000], is the distance measure used for this purpose [Aucouturier and Pachet, 2004, Pampalk, 2006]. The EMD algorithm finds the minimum work required to transform one distribution into another.

[2] `http://www.allmusic.com/`
[3] `http://www.lastfm.com/`

2.4 Bringing It Together

There has been some recent work attempting to bridge the divide between content-based analysis and human generated metadata. Most of this work [Knees et al., 2006, Slaney and White, 2007] focuses on various ways of exploiting the human-generated metadata to filter content prior to, or instead of, conducting content-based analysis, similar to the techniques discussed in Section 2.3, in order to reduce computational load. The research detailed in Chapter 3 begins to explore this space more fully than has been seen in previous works. This is done by working to exploit both relational data and content based features in parallel [Fields et al., 2008a,b].

Chapter 3

Current Work

In order to best understand the creation of playlists the best methods for harnessing information for two separate domains are explored. Section 3.1 discusses methods for acquiring and preparing data from Myspace social network's artist network. Section 3.2 discusses work examining the relationship between the social distance of artist nodes and how this distance corresponds to acoustic distance. Section 3.3 looks into how another graph metric, maximum flow analysis, relates to acoustic data. Both of these methods outline experimental results and preliminary means for producing playlists using each.

3.1 Data Acquistion

3.1.1 Sampling the Social Web

The Myspace social network presents a variety of challenges. For one, the size of the network prohibits analyzing the graph in its entirety, even when considering only the artist pages. Therefore the current work deals with a sample (of large absolute size) of the network. Also, the Myspace social network is filled with noisy data – plagued by spammers and orphaned accounts. The scope of the sampling is limited in a way that minimizes this noise. And finally, there currently is no interface for easily collecting the network data from Myspace.Our data is collected using web crawling and HTML scraping techniques[1].

Artist Pages

It is important to note this work only concerns a subset of the Myspace social network – the Myspace *artist* network. Myspace artist pages are different from standard Myspace pages in that they include a distinct audio player application. The presence or absence of this player is used to determine whether or not a given page is an artist page.

A Myspace page will most often include a top friends list. This is a hyperlinked list of other Myspace accounts explicitly specified by the user. The top friends list is limited in length with a maximum length of 40 friends (the default length is 16 friends). In constructing our sampled artist network, the top friends list is used to create a set of

[1]Myspace scraping is done using tools from the MyPySpace project available at `http://mypyspace.sorceforge.net`

directed edges between artists. Only top friends who also have artist pages are added to the sampled network; standard Myspace pages are ignored. The remainder of the friends list is also ignored(*i.e.* friends that are not specified by the user as top friends), assuming these relationships are not as relevant. This reduces the amount of noise in the sampled network but also artificially limits the outdegree of each node. Our sampling is based on the assumption that artists specified as top friends have some meaningful musical connection for the user – whether through collaboration, stylistic similarity, friendship, or artistic influence.

The audio files associated with each artist page in the sampled network are also collected for feature extraction. Cached versions of the audio files are downloaded and audio features are extracted.

Snowball Sampling

There are several network sampling methods; however, for the Myspace artist network, snowball sampling is the most appropriate method [Ahn et al., 2007, Lee et al., 2006]. In this method, the sample begins with a seed node (artist page), then the seed node's neighbors (top friends), then the neighbors' neighbors, are added to the sample. This breadth-first sampling is continued until a particular sampling ratio is achieved. Here, an artist is randomly selected[2] and used to seed the crawler. The crawler collected all artist nodes within 6 edges of the seed node to a total of 15,478 nodes. This produces a dataset where no more than six directed top friends links need to be followed to get from the seed artist to any other artist in the dataset. If the size of the Myspace artist network is around 7 million[3], then this dataset approximates the 0.25% sampling ratio suggested for accurate degree distribution estimation in sampled networks. However, it is insufficient for estimating other topological metrics such as the clustering coefficient and assortativity [Kwak et al., 2006]. Of course, a complete network topology is not the primary concern here. This crawling method can be seen in Figure 3.1.

With snowball sampling there is a tendency to over sample hubs because they have high indegree connectivity and are therefore picked up disproportionately frequently in the breadth-first sampling. This property would reduce the degree distribution exponent γ and produce a heavier tail but preserve the power-law nature of the network [Lee et al., 2006].

3.1.2 Network Analysis

The Myspace artist network sample exhibits many of the network characteristics common to social networks and other real-world networks. Some of the network statistics are summarized in Table 3.1.

Although the network is constructed as a directed network, it is treated as an undirected network to simplify analysis. Each edge is considered bi-directional, that is $(i, j) = (j, i)$, and if a reflexive pair of edges existed in the directed graph, only one

[2]The artist is *Karna Zoo*, Myspace url: `http://www.myspace.com/index.cfm?fuseaction=user.viewProfile&friendID=134901208`

[3]`http://scottelkin.com/archive/2007/05/11/MySpace-Statistics.aspx` reports as of April 2007 ~25 million songs, our estimates approximate 3.5 songs/artist, giving ~7 million artists

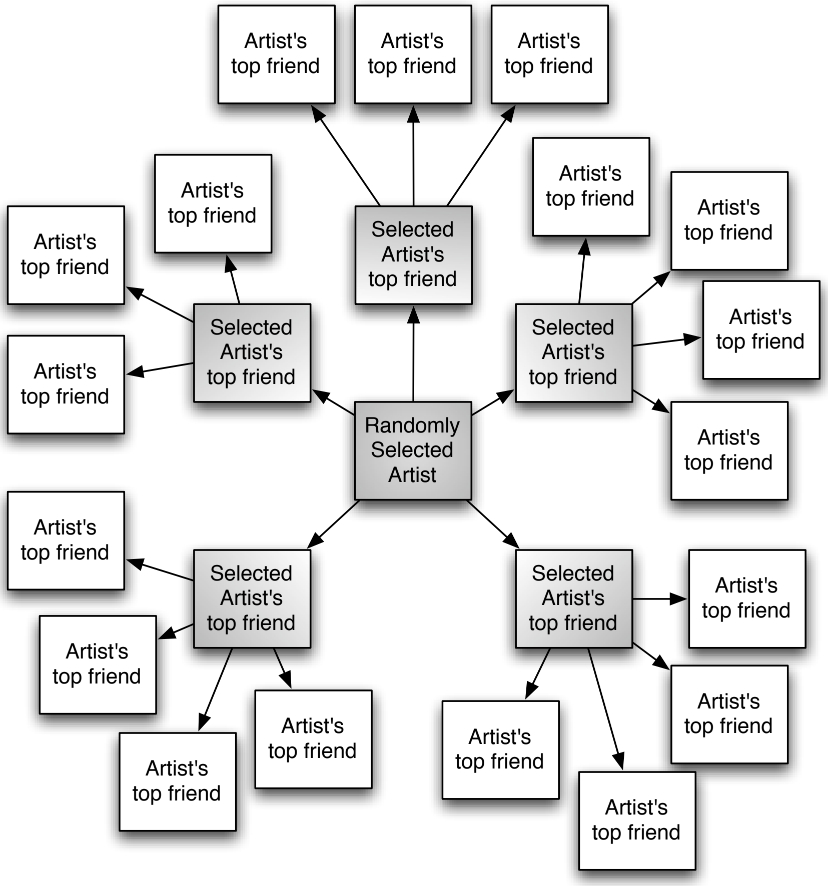


Figure 3.1: A representative graph of a snowball sample of a complex network. Note that while this simplified graph contains no cycles, cycles can and do exist in real world complex networks, including the Myspace artist network.

bi-directional edge exists in the undirected graph. An examination of the directed graph is reserved for later work.

The degree distribution for the undirected network is plotted in Figure 3.2 on a log-log scale. As mentioned earlier, it is common to find a power-law degree distribution in social networks [Newman, 2003]. However, exponential degree distributions have been reported previously in some types of music recommendation networks [Cano et al., 2006]. This is especially true for networks with imposed degree limits. For moderate degree values ($35 < k < 200$), our sample shows a power-law distribution. For lower degree values, the distribution is closer to exponential. This may be related to the fact that our network has an out degree limit imposed by Myspace restricting the maximum number of top friends ($k_{out} \leq 40$). The power-law fit also breaks down for high values of k – most likely due to the limited scope of our sample. Similar "broad-scale" degree distributions have been reported for citation networks and movie actor networks [Amaral et al., 2000].

3.1.3 Signal-based analysis

MFCCs are extracted from each audio signal using a Hamming window on 8192 sample FFT windows with 4096 sample overlap. All MFCCs are created with the fftExtract

	n	m	$\langle k \rangle$	l	d_{max}
undirected	15478	91326	11.801	4.479	9
directed	15478	120487	15.569	6.426	16

Table 3.1: The network statistics for the Myspace artist network sample where n is the number of nodes, m is the number of edges, $\langle k \rangle$ is the average degree, l is the mean geodesic distance, and d_{max} is the diameter, as defined in Section 2.2.1.

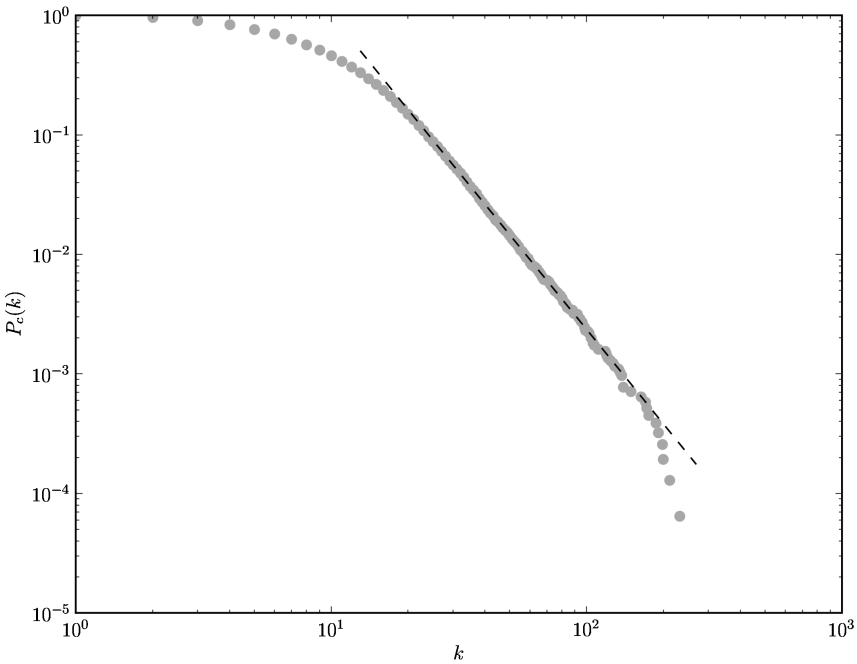


Figure 3.2: The cumulative degree distributions for the Myspace artist network sample. For moderate values of k, the distribution follows a power-law (indicated by the dotted line), but for low and high values the decay is exponential.

tool[4]. For each artist node a GMM is built from the concatenation of MFCC frames for all songs found on each artist's Myspace page. Generally artists have between 1 and 4 songs, although some artists have many more. The mean number of songs is slightly more that 3.5 per artist. An $n \times n$ matrix is populated with the earth mover's distance λ_{ij} between the GMMs corresponding to each pair of nodes in the sample.

3.2 Geodesic Distance

To begin, the structure of the Myspace artist network sample is analyzed – showing that it conforms, in most respects, to the topology expected from such a social network. A simple metric is developed for exploring the interaction between signal-based music similarity and the network structure.

[4]source code at http://omras2.doc.gold.ac.uk/software/fftextract/

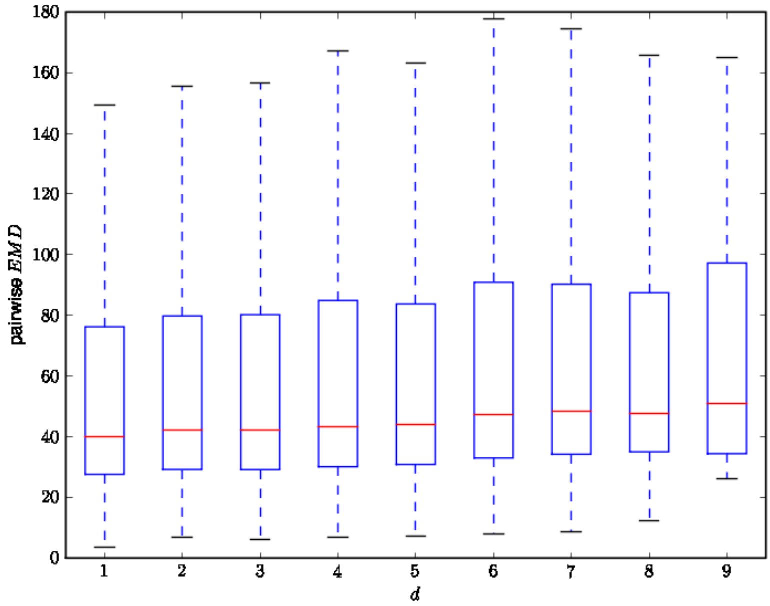


Figure 3.3: The box and whisker plot showing the spread of pair-wise artist dissimilarity grouped by geodesic distance as found on the artist graph.

3.2.1 Relationship with signal-based measures

A simple relation between audio signal dissimilarity and network structure is explored using a box and whisker plot. The plot is shown in Figure 3.3. For all pairs of artists i and j, the EMD dissimilarity is found(λ_{ij}). These dissimilarities are grouped according to the geodesic distance in the undirected network between the artist nodes i and j, d_{ij}. There appears to be no clear correlation between these λ values and geodesic distance. The Pearson product-moment correlation coefficient confirms this giving a ρ of -0.0016, with a p value of 1.50×10^{-20}. This should be viewed in the context of the number of pairwise relationships used, implying it is stable, at least for the community of artists found via this sample of the network.

3.2.2 Geodesic Playlisting

In order to extend this cross domain analysis into playlist generation, a weighted directed graph is used. This graph is the sampled graph as described in Section 3.1.1 with weights added to each edge describing the acoustic dissimilarity between each of the artists joined by a given edge. A given edge E_{ij} has as its weight an acoustic distance λ_{ij} as described in Section 3.2.1. An example of such a graph can be seen in Figure 3.4. Once this weighted directed graph is constructed, ordered artist lists are produced by finding the geodesic distance from a starting artist to a destination artist. An example path of this sort can be seen in Figure 3.5

It is important to note that this method does not give playlists per say. Rather, it produces orders lists of related artists. It remains a further task to select one of each artists' songs in order to produce the final ordered list of songs. This can be done

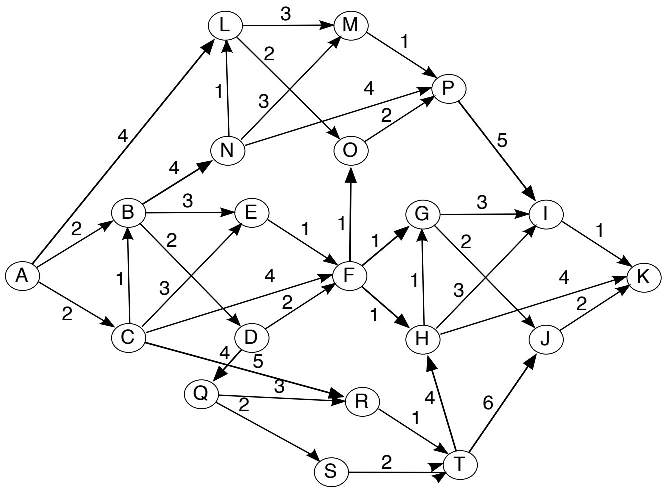


Figure 3.4: A weighted directed graph, similar to one that would be used to generate geodesic playlists.

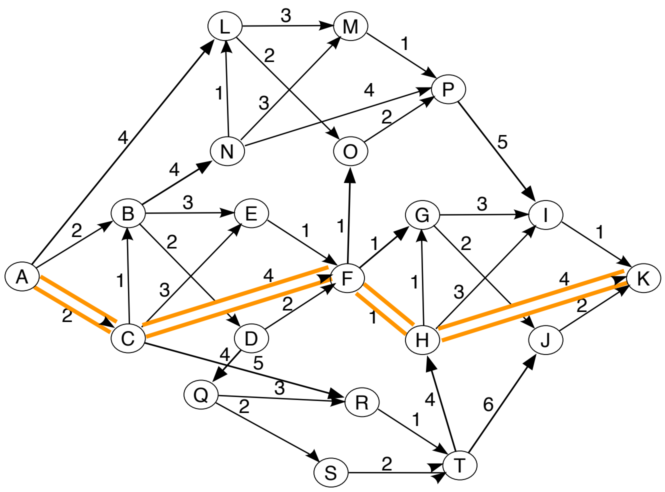


Figure 3.5: A representative geodesic path from node A to node K giving an order list of nodes (artists) representing an artist list used in geodesic playlisting.

by selecting each artists' most popular track or employing any other sensible metric to determine the most representative song for that artist.

3.3 Minimum Cut/Maximum Flow

The Maximum Flow value is used as a means of determining the number of independent paths from a source node to a sink node. Formally the Maximum Flow/Minimum Cut theorem [Elias et al., Dec 1956], it is used to calculate the highest weight in the narrowest part of the path from source to sink. The theorem's name comes from the equivalence in the smallest weight of edges that must be removed in order to create two subgraphs which disconnect the source and sink nodes. Further, if the edges in the graph are unweighted, this value is also equivalent to the number of paths from the source to the sink which share no common edges. As this is a mature algorithm there are a number of optimization strategies that have been examined [Ahuja et al., 1993, Goldberg and Tarjan, 1988].

An example of Maximum Flow can be seen on the network in figure 2.1. It can been seen that the narrowest point from node A to node F is $E_{a,b}$ and $E_{a,c}$. The maximum flow can simply be found via Equation 3.1.

$$M = \sum m(i,j) \tag{3.1}$$

Where $m(i,j)$ is the magnitude of each edge in the minimum cut set.

In our Myspace top friends graph, the maximum flow is measured on the unweighted directed graph from the source artist node to the sink artist node.

3.3.1 Experiment

The maximum flow value is calculated, using the snowball sample entry point as the fixed source against every other node in turn as a sink, yielding the number of edges connecting each sink node to the entry point node at the narrowest point of connection. The acoustic distances can then be compared to these maximum flow values.

Signal-based analysis

The acoustic distances used are calculated via the same means described in Section 3.1.3, yielding a gaussian mixture model describing the average of all the artist's song content. These GMMs are then compared via the earth mover's distance to find pairwise dissimilarity figures.

Random Networks

In order to better understand a result from analysis of our Myspace sample, a baseline for comparison must be used. To that end, random permutations of the node locations are examined. In order to preserve the overall topology present in the network, this randomization is performed by randomizing the artist label and associated music attached to a given node on the network. This is done ten fold, creating a solid baseline to test the null hypothesis that the underlining community structure is not responsible for any correlation between maximum flow values and Earth Mover's Distance.

3.3.2 Result

The results of the first experiment show no simple relationship between the sampled netwrok and the randomized network. This can be seen in Table 3.2 and in Figures 3.6a and 3.6b. There is an increase in the median EMD for the less well connected (*i.e.* lower maximum flow value) node pairs in the Myspace sample graph, though this is not significant enough to indicate a correlation, while the randomized permutations are near flat. While the median EMD of the artist pairs with a maximum flow of 10 is appreciably higher than all other in the randomized graph, this is likely related to the relatively large size of this group. Perhaps the easiest way to examine the relationship between the sampled graph and randomized one is through the deltas of each group's median from the entire dataset median. This data is shown in the second and forth colum in Table 3.2 and Figure 3.7. Further, the Kruskal-Wallis one-way ANOVA results for both the sample graph and averaged across the 10 fold permutations are shown in Table 3.3.

3.3.3 The Max Flow Playlist

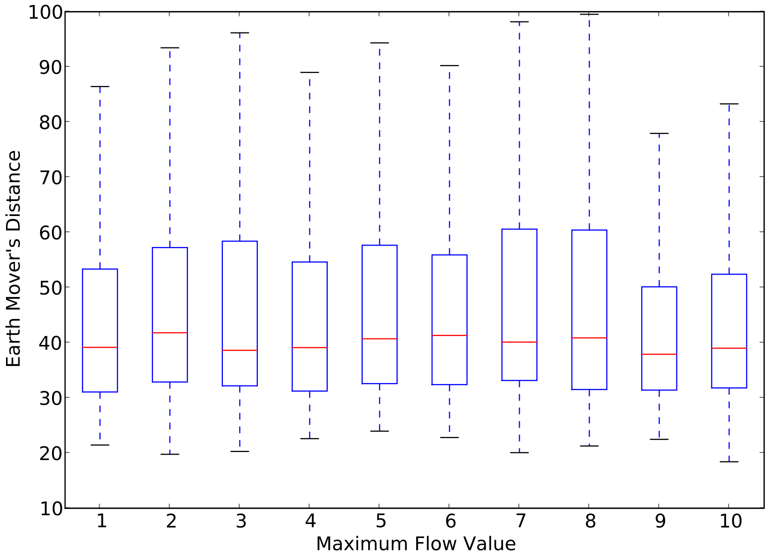
In order to build playlists using both acoustic and social network data, the Earth Mover's Distance is used between each pair of neighbors as weights on the Myspace sample network. Two artists are then selected, a starting artist as the source node and a final artist as the sink node. One or more paths are then found through the graph via the maximum flow value, generating the list and order of artists for the playlist. The song used for

Max Flow	median	deviation	randomized	deviation
1	40.80	1.26	39.10	−0.43
2	45.30	5.76	38.34	−1.19
3	38.18	−1.35	38.87	−0.66
4	38.21	−1.32	38.64	−0.89
5	40.00	0.47	39.11	−0.42
6	41.77	2.25	39.02	−0.51
7	39.94	0.41	39.24	−0.29
8	39.38	−0.15	38.76	−0.77
9	38.50	−1.03	38.87	−0.66
10	39.07	−0.46	40.85	1.32

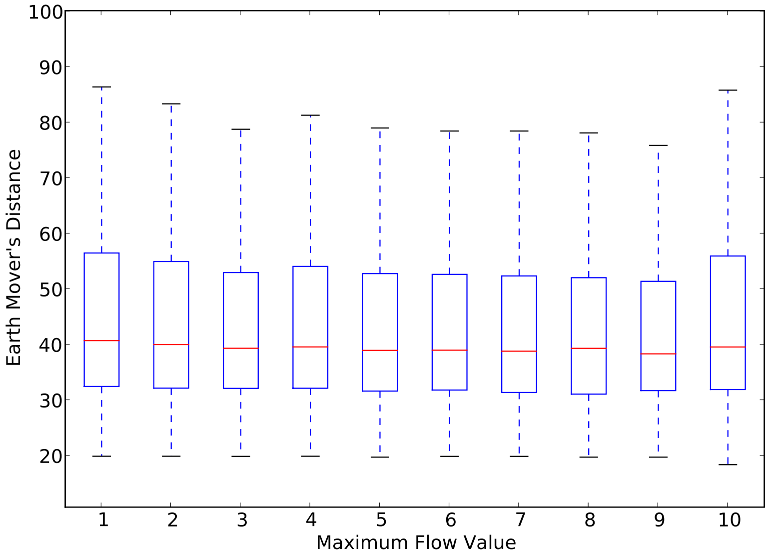
Table 3.2: Node pairs average EMD values grouped by actual minimum cut values and randomized minimum cut values, shown with deviations from the global median of 39.53.

	H-value	P-value
From sample	12.46	0.19
Random permutations	9.11	0.43

Table 3.3: The Kruskal-Wallis one-way ANOVA test results of EMD against maximum flow for both the sampled graph and it's random permutations. The H-values are drawn from a chi-square distribution with 10 degrees of freedom.



(a) The EMD distribution on the sampled graph



(b) The EMD distribution on the random permutations of the graph, maintaining the original edge structure.

Figure 3.6: The box and whisker plots showing the distribution of EMD grouped by maximum flow value between artists on the Myspace social graph and the randomized permutations of the graph.

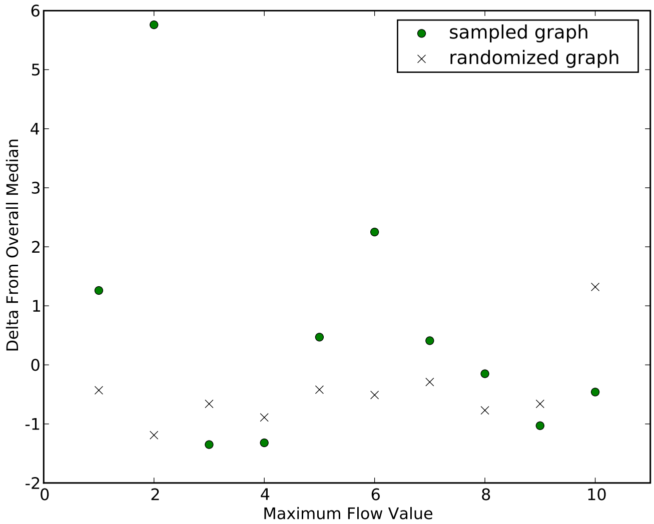


Figure 3.7: The deltas from the global median for each maximum flow value group of EMD values, from the sampled graph and the randomized graph.

each artist is the most popular at the time of the page scrape. In this way playlists are constructed that are both influenced by timbre similarity and bound by social context, regardless of any relationship found between these two spaces found via the work discussed in Section 3.3.1. Playlists generated using this technique were informally auditioned, and were found to be reasonable on that basis.

Chapter 4

Going Forward

The results of the two experiments detailed in Chapter 3 were decidedly mixed. However, while not perfectly orthogonal, the artist social graph and the acoustic dissimilarity matrix clearly encode different relational aspects between artists. This suggests that a recommendation system that can use both domains well has the potential to perform better than a similar system that relies on only one domain in isolation.

In Section 4.1 the particulars of continuing ahead with the previous work on the Myspace dataset are explained. This includes the proposal of a deployable real world playlist generating system in Section 4.1.3. In Section 4.2 further experiments are proposed looking at the creation of playlists using content -based features and trained models based on examples. Last, Section 4.3 will look into evaluation, including possible methods, anticipated problems and other related issues.

4.1 Extension and Completion of Myspace Dataset

The dataset captured from Myspace as detailed in Section 3.1 has proven to be an excellent means of exploring the relationship between social data and acoustic data. In addition to the analysis described in Chapter 3 this dataset has been used to do novel work examining the organization of communities of artists, both with and without the accompanying acoustic data [Jacobson et al., 2008, Jacobson and Sandler, 2008]. Additionally, rdf translation services are being used as part of the rdf open data network[1].

4.1.1 Toward a Complete Capture

One of the next pieces of work that is being done with the Myspace data set is to continue the crawling process in an effort to capture a lager sampled set that approaches the complete artist subnetwork of Myspace. While this may seem a straightforward extension of previous data capture efforts, the scaling up the crawler is non-trivial for a number of reasons. Acknowledging that a complete capture of the artist may be out of reach given current constraints, there are many ways to significantly improve the

[1] The service, which generates the same rdf that is generated by the crawler, is available at `http://dbtune.org/myspace`. At the time of writing at least one web services company (`http://www.garlik.com`) is known to be using this service.

representation of the captured subset, perhaps to the point that a complete capture is not statistically necessary.

Problems of Scale

The most severe bottleneck preventing a complete capture of the network is the computational load required in order to extract features for all of the audio associated with each artist. Based on estimates from the samples currently captured and the approximate size of the network, this would take years of continuous processing given the computational power available. Even if sufficient cycles were available to reduce the compute time of feature extraction for the whole network, the crawl itself is also prohibitively slow in achieving a crawl over the network of the order of 10^8 nodes and its associated media. This latter time could potentially be decreased to an acceptable level if and when Myspace implements a functional version of their web service API[2].

Mechanisms for Improving the Captured Subset

It may not be necessary to capture the entire dataset in order to solve most of the topology based problems that exist in the current dataset. The crawl now works from multiple randomly selected seed artists. Each seed artist will be used to start a snowball sample until that sample has joined the larger graph or is complete. With a sufficient number of artist nodes ($10^6 - 10^7$) a much more representative sample will have been gathered would should be sufficient to continue to conduct further experiments with a more robust dataset.

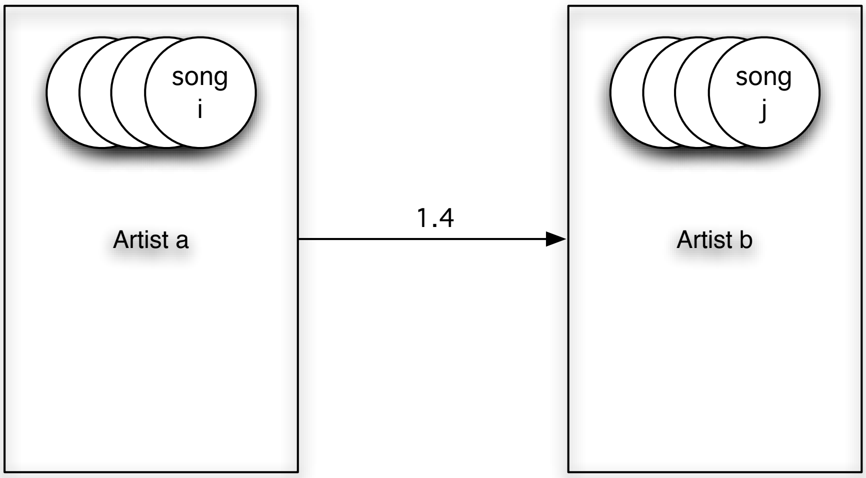
4.1.2 Extensions of Complexity

In order to move beyond the creation of artist lists as described in Sections 3.2.2 and 3.3.3 to directly creating playlists, a further complexity must be introduced into the graph. Rather than have a node for each artist, as the current graph is constructed, the basic node will be changed to be a song. The edge connections will reflect the social connections between artists associated with the song nodes. The weight of an edge E_{ij} will be the acoustic dissimilarity between song i and song j. These node relationships can be seen in Figure 4.1b. The acoustic dissimilarity will remain as described in Section 3.2.1, though it may be changed in favor of a less complex or more accurate method if one is found to be experimentally viable for these tasks.

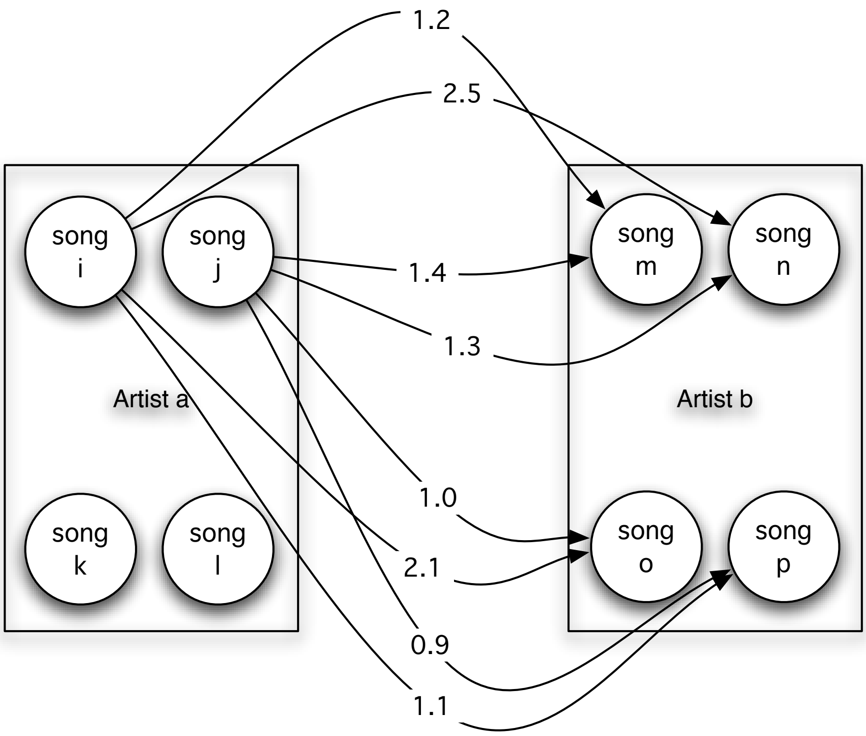
4.1.3 Democratic Social Radio

Once the means of graph representation has been extended to directly provide songs as output, the following system is proposed as a means of deployment. This system will be designed to play a continuous stream of songs via the Goldsmiths student radio stations internet radio stream, while the station is not usually broadcasting (typically from midnight till 6am, sometimes 10pm till 10am). The playback system will begin

[2]Myspace has publicly committed to joining the OpenSocial (`http://code.google.com/apis/opensocial/`) inititive which would provide web serviceable entry points to greatly speed up the crawler.



(a) The original artist to artist relationship



(b) The proposed new artist relationship, with songs as nodes. Note that the connections of song k and song l have been omitted for clarity.

Figure 4.1: A comparison of current and proposed future means of representing the relationship between artists.

with an initial seed song and destination song, then construct a playlist. While this playlist is being broadcast, anyone tuning into the broadcast will also being able to vote via a web based application on the next song to serve as the destination. Once the current destination song begins broadcast, the voting for the next cycle will cease. The current destination song will be considered the seed song for the next cycle and the song with a plurality of votes will become the new destination, then the next playlist will be calculated and its members broadcast. This process will continue until regular human based broadcast resumes in the morning.

If this automatic playlist creation system is allowed to run for a sufficient amount of time, a great deal of user data will be recorded. This would include voting behavior, average length of time continuously listened and whether listeners (or at least IP addresses) return.

4.2 Playlist Creation

In addition to continuing to move forward with the Myspace dataset, there will also be more detailed exploration into generating playlists through content based features. This will involve improving the state of the art in how playlists are specified by a user and better understanding and modeling the balance between novelty and accessibility as well as the related ideas of expectation and surprise.

4.2.1 More Specific Queries

As the qualitative results show in Flexer et al. [2008], playlist results are vastly improved by the additional information provided by the a second song being included in a query system. Although this is not the only method of further query specification acknowledged in literature, the critical point at this stage is that any viable novel system must have a means for the user to specify not only a starting position but the trajectory as well.

4.2.2 Novelty Curves and Expectation

Built into ideas of any ordered list of stimuli are changing perceptions of novelty and expectation. In order to make better playlists, if relying on content based features, this information of a novelty curve must be encoded somewhere else. A statistical model system similar to that described in Platt et al. [2002] could be extended to work with extracted audio features rather than textual features. In order to encapsulate novelty playlists from Ellingham [2007] and Lynskey [2008] will be used as training data. The goal of this approach is to apply the dynamic shape of aggregate expert playlists onto new material.

4.2.3 Automated Mixing

In order to make a complete automated playlist broadcasting system, some improvement is needed to the current standard automatic crossfade. Some amount of phrase segmentation and alignment of neighboring tracks on certain types of playlists has the potential to vastly improve subjective response to computationally generated playlists' automatic playback. This system could also involve content responsive equalization, though this idea has very little literature behind it and may prove extremely difficult.

4.3 Evaluation

Evaluation is a tricky issue within any form of playlist generation. This comes principally from two related factors. The first factor is that the problem of creating a *correct* playlist is a very vaguely defined problem. This can be mitigated through specificity. In each of the two proposed playlist generation systems a level of specificity is inherent in the problem that will ease evaluation. In the broadcast playlist system proposed in Section 4.1.3 correctness is defined as listener satisfaction and retention. These will be assessable based upon recorded aggregate user data once the system has been live for a sufficient length of time. The goal of the content based playlist system outlined in Section 4.2 is not to produce exact song results based upon a preordained metric but rather to produce playlists that conform to specific song to song dynamic shapes. While this is certainly a less well defined problem than that of the social radio, it remains testable.

The second evaluation problem is one of scope. As you add context to a recommender system in any domain, you add information. In the domain of music, this is seen in the large length of time required to listen to many iterations of a playlist generators output. This makes traditional listening tests generally time prohibitive. The way this will be dealt with in the broadcast playlist is basically self-evident. By simply turning the system on and recording user activity, users can provide passive listening test data simply through use. The noise this may introduce into the results can to a great extent be mitigated by allowing the system to run until the volume of collected user data balances out the inevitable noise of the collection method. Dealing with this in the content-based modeling system is a bit more tricky, if it is determined that exhaustive listening tests are required. There are some very promising means of evaluating these sorts of playlists that sidestep the listening test problem as they provide a means to computationally determine correctness. The most widely used is to evaluate the likelihood of genre change between members on the list. However this metric may not be applicable in this case, as it cannot be taken for granted, given the goal of emulating novelty/expectaction curves from example, that genre is or is not desirable. The best avenue for a computational approach to evaluation for this playlist generation method may be in comparing the novelty curve of each generated playlist against some ideal curve. Clearly the means for evaluating this particular generation method is no yet solved and somewhat open to research.

Research Plan

Date of Completion	**Task**
February 2009	Finish and submit journal article detailing current state of the art in playlist generation and aesthetics (targeting either CMJ or JNMR)
February/March 2009	Go live with first revision of social radio service and begin data collection
March 2009	Evaluate status of content aware song to song mixing. If viable submit to DaFX09
April 2009	data gathered for playlist curve training
May 2009	complete ISMIR 2009 paper detailing social radio service and preliminary analysis of collected user data
Late May 2009	Go live with a second version of social radio service to run over the summer
June 2009	run first pass of content based playlist generation system
August 2009	pending task approval, enter MIREX2009 playlist generation system with content based system improved from previous runs.
August 2009	Evaluate aggregate data from extended social radio service run
January 2010	Finish all major research work and begin writing thesis
May 2010	Submit completed PhD Thesis
June 2010	PhD Viva

Bibliography

Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th international conference on World Wide Web*, Alberta, Canada, May 2007. IW3C2.

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-617549-X.

M. Alghoniemy and A. Tewfik. A network flow model for playlist generation. In *Multimedia and Expo, 2001. IEEE International Conference on*, 2001. URL `citeseer.ist.psu.edu/alghoniemy01network.html`.

L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley. Classes of small-world networks. In *Proceeding of the National Academy of Sciences*, 2000.

S. S. Anand and B. Mobasher. *Contextual Recommendation*, pages 142–160. Springer Berlin / Heideberg, 2007.

A. Anglade, M. Tiemann, and F. Vignoli. Virtual communities for creating shared music channels. In *Proc. of Int. Symposium on Music Information Retrieval*, 2007.

J.-J. Aucoutier and F. Pachet. Scaling up playlist generation. In *Proc. IEEE International Conference on Multimedia and Expo*, 2002.

J. Aucouturier and F. Pachet. Improving timbre similarity : How high's the sky ? *J. Negative Results in Speech and Audio Sciences*, 2004.

P. Avesani, P. Massa, M. Nori, and A. Susi. *Collaborative Radio Community*, pages 462–465. Springer Berlin / Heideberg, January 2002. doi: 10.1007/3-540-47952-X.

A. Berenzweig, B. Logan, D. P. W. Ellis, and B. P. W. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music J.*, 28(2):63–76, 2004. ISSN 0148-9267. doi: http://dx.doi.org/10.1162/014892604323112257.

B. Bewster and F. Broughton. *Last Night A DJ Saved My Life; The history of the disc jockey.* Headline Book Publishing, London, United Kingdom, 2nd edition, 2006.

P. Cano, O. Celma, M. Koppenberger, and J. M. Buldu. The topology of music recommendation networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2006. URL `http://www.citebase.org/abstract?id=oai:arXiv.org:physics/0512266`. http://arxiv.org/abs/physics/0512266v1.

W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. *ACM Trans. Inf. Syst.*, 17(2):141–173, 1999. ISSN 1046-8188. doi: http://doi.acm.org/10.1145/306686.306688.

L. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas. Characterization of complex networks: A survey of measurements. *Advances In Physics*, 56:167, 2007. URL `doi:10.1080/00018730601170527`.

M. Dunlop and A. Crossan. Predictive text entry methods for mobile phones. In *Personal and Ubiqutous Computing*, volume 4, pages 134–143. Springe London, June 2000. doi: 10.1007/BF01324120.

P. Elias, A. Feinstein, and C. Shannon. A note on the maximum flow through a network. *Information Theory, IEEE Transactions on*, 2(4):117–119, Dec 1956. ISSN 0018-9448.

M. Ellingham, editor. *The Rough Guide book of Playlists*. Rough Guides Ltd., 2nd edition, 2007.

B. Fields, K. Jacobson, M. Casey, and M. Sandler. Do you sound like your friends? exploring artist similarity via artist social network relationships and audio signal processing. In *Int. Computer Music Conference*, August 2008a.

B. Fields, K. Jacobson, C. Rhodes, and M. Casey. Social playlists and bottleneck measurements : Exploiting musician social graphs using content-based dissimilarity and pairwise maximum flow values. In *Proc. of Int. Symposium on Music Information Retrieval*, September 2008b.

A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist generation using start and end songs. In *Proc. of Int. Symposium on Music Information Retrieval*, October 2008.

P. Gleiser and L. Danon. Community structure in jazz. *Advances in Complex Systems*, 6 (4):565–573, 2003.

A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, 1988. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/48014.61051.

C. Hayes and P. Cunningham. Context boosting collaborative recommendations. *Knowledge-Based Systems*, 17(2-4):131–138, 2004. ISSN 0950-7051. doi: DOI:10.1016/j.knosys.2004.03.008. URL `http://www.sciencedirect.com/science/article/B6V0P-4C2PYHT-5/2/5091bc5f7ae1f7668a1f27a3239e8f50`. AI 2003, the Twenty-third SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence.

K. Jacobson and M. Sandler. Musically meaningful or just noise, an analysis of on-line artist networks. In *Proc. of CMMR*, pages 306–314, 2008.

K. Jacobson, B. Fields, and M. Sandler. Using audio analysis and network structure to identify communities in on-line social networks of artists. In *Proc. of Int. Symposium on Music Information Retrieval*, 2008.

P. Knees, T. Pohle, M. Schedl, and G. Widmer. Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *Proc. 8th ACM international workshop on Multimedia information retrieval*, pages 147 – 154, 2006.

H. Kwak, S. Han, Y.-Y. Ahn, S. Moon, and H. Jeong. Impact of snowball sampling ratios on network characteristics estimation: A case study of cyworld. Technical Report CS/TR-2006-262, KAIST, November 2006.

R. Lambiotte and M. Ausloos. On the genre-fication of music: a percolation approach (long version). *The European Physical Journal B*, 50:183, 2006. URL `doi:10.1140/epjb/e2006-00115-0`.

J. Lanza. *Elevator Music: A Surreal History of Muzak, Easy-listening, and Other Moodsong*. University of Michigan Press, 2nd edition, 2004.

S. H. Lee, P.-J. Kim, and H. Jeong. Statistical properties of sampled networks. *Physical Review E*, 73:102–109, January 2006.

B. Logan. Mel frequency cepstral coefficients for music modeling. In *Int. Symposium on Music Information Retrieval*, 2000.

B. Logan and A. Salomon. A music similarity function based on signal analysis. *Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on*, pages 745–748, 22-25 Aug. 2001.

D. Lynskey. *Book of Playlists; The Best of the Guardian's 'Readers Recommended'*. Aurum Press in association with Guardian Books, 2008.

H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM J. Discret. Math.*, 5(1):54–66, 1992. ISSN 0895-4801. doi: http://dx.doi.org/10.1137/0405004.

M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167, 2003. URL `http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0303516`.

E. Pampalk. *Computational Models of Music Similarity and their Application in Music Information Retrival*. PhD thesis, Technischen Universität Wien, May 2006.

E. Pampalk, T. Pohle, and G. Widmer. Dynamic playlist generation based on skipping behavior. In *Proc. of Int. Symposium on Music Information Retrieval*, 2005.

J. Park, O. Celma, M. Koppenberger, P. Cano, and J. M. Buldu. The social network of contemporary popular musicians. *Int. J. of Bifurcation and Chaos*, 17:2281–2288, 2007. URL `http://www.citebase.org/abstract?id=oai:arXiv.org:physics/0609229`.

J. C. Platt, C. J. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a gaussian process prior for automatically generating music playlists. In *Proc. Advances in Neural Information Processing Systems*, volume 14, pages 1425–1432, 2002.

R. Ragno, C. Burges, and C. Herley. Inferring similarity between music objects with application to playlist generation. In *Proc. 7th ACM SIGMM international workshop on Multimedia information retrieva*, 2005.

Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

M. Slaney and W. White. Similarity based on rating data. In *Proc. of Int. Symposium on Music Information Retrieval*, 2007.

T. Stocky, A. Faaborg, and H. Lieberman. A commonsense approach to predictive text entry. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1163–1166, New York, NY, USA, 2004. ACM. ISBN 1-58113-703-6. doi: http://doi.acm.org/10.1145/985921.986014.