Modeling Musical Influence Through Data

Modeling Musical Influence Through Data

Abstract

Musical influence is a topic of interest and debate among critics, historians, and general listeners alike, yet to date there has been limited work done to tackle the subject in a quantitative way. In this thesis, we address the problem of modeling musical influence using a dataset of 143,625 audio files and a ground truth expert-curated network graph of artist-to-artist influence consisting of 16,704 artists scraped from AllMusic.com. We explore two audio content-based approaches to modeling influence: first, we take a topic modeling approach, specifically using the Document Influence Model (DIM) to infer artist-level influence on the evolution of musical topics. We find the artist influence measure derived from this model to correlate with the ground truth graph of artist influence. Second, we propose an approach for classifying artist-to-artist influence using siamese convolutional neural networks trained on mel-spectrogram representations of song audio. We find that this approach is promising, achieving an accuracy of 0.7 on a validation set, and we propose an algorithm using our trained siamese network model to rank influences.

Contents

List of figures

List of Tables

To my family.

Acknowledgments

Here's to my own influences!

First of all, I would like to thank my advisors, Mark Glickman and Yaron Singer for putting up with me taking on this passion project for my senior thesis. Besides being brilliant people, you are also two of the most genuine human beings I have met during my time at Harvard. In all likelihood, in 20 years all three of us probably will have forgotten about this thesis, but I will never forget your kindness towards me.

I thank James Waldo for graciously agreeing to being a reader for this thesis.

I thank Raghu Dhara, Mark Goldstein and Kojin Oshiba for their invaluable advice in helping me brainstorm ideas for this thesis.

I would like to thank my suitemates: Chris Chen, Josh Felizardo, Phillip Huang and Mario Menendez. True friends are hard to find in this world; hopefully this means that you'll still talk to me after graduation.

I thank Dan Fox, Jeff Harrington, Mark Kaplan, Mark Olson, Paul Pitts, Peter Tileston and the many other musical mentors I have had over the years who helped instill an undying love for music in me. I've learned so much from music beyond just the notes, and it's in no small part due to you.

To Jasper Schilling, one of the few people that I will admit is perhaps a bigger music fan than me. Thanks for being a friend all these years, and I wish you the best as you embark on your own career in music.

I thank my little sister, Weian for always being adorable. Enough said.

Finally, I have to thank my parents, whom I really can't thank enough. I know that I may not always show it, but I truly love you with all my heart and am grateful for all the sacrifices you have made for me over the years. *Ma,* thank you for always being a great listener and for your indefatigable love. *Ba,* thank you for showing me by example how to be my own man. My time in college has not always been smooth sailing, but you two helped me see it through. Thank you. This is for you.

Then reach the stars, you take the time
To look behind and say, "Look where I came
Look how far I done came"

"Intro"- J. Cole

1

Introduction

1.1 Motivation

The study of musical influence relationships is a topic of great interest to music researchers, critics and general enthusiasts alike. As humans, we often use influence terminology in order to situate a musical artist on the sonic spectrum. For instance, music critics will often introduce a new artist in terms of their influences. As another example, a person recommending an artist to a friend will often speak of the musician in terms of *who* he or she sounds like.

Though pretty intuitive to humans, modeling influence computationally is not a straightforward endeavor. Part of the difficulty in modeling musical influence computationally arises from a lack of a precise definition for influence, making it a rather abstract task. Though most people have a good general sense of what it means for one musical artist to have influenced another, in reality influence can take on several different meanings. As Morton and Kim note [14], one artist may have exerted direct influence on another artist through direct and prolonged personal interactions. These interactions include "teacher-student relationships, band membership, frequent collaborations between artists, and even familial associations" [14]. Not all influence relationships take on this sort of flavor however; many artists are voracious listeners themselves and will be influenced by something as brief as a 10 second segment of a song that they happen to hear by chance while sitting in a coffee shop. The genre of hip-hop is a salient example of this phenomenon, as producers and musicians will often use such "found sounds" as samples that are incorporated in the creation of new works.

Influence also varies in the way it ultimately manifests itself in an artist's work. In the case of hip-hop, sometimes this can be rather obvious as an artist will directly sample a strong influence of theirs. In other instances however, detecting influential elements can be much more difficult. For example, a jazz musician might try to incorporate minutia such as the articulation patterns, harmonic vocabulary and/or timbre of an influence in their own playing. Such influence can be difficult to detect for even the human listener, requiring a keen ear and extended listening to unpack. Furthermore, it is also important to note the distinction between influence and similarity. Though artists who exhibit an influence relationship often will sound similar, this is not necessarily the case. As Morton and Kim point out, "one artist may have had a large influence on one another and yet the two musicians differ greatly in terms of perception" [14].

In addition to the challenge of a lack of a clear definition for influence, the problem of modeling influence also reflects broader challenges in the field of music information retrieval (MIR) in general, especially if one is to take an audio-based approach. First, audio data is quite complex, containing rich structural information on multiple timescales and second, music itself is ever-evolving as artists, songs and genres all change over time [18]. With regard to dealing with the complexity of audio data, there still exists a large semantic gap in extracting high-level properties such as "genre, mood, instrumentation and themes" from audio [20]. In terms of the evolving nature of music, it is quite difficult to create models complex enough to capture these shifting relationships.

Despite these challenges, inferring musical influence surely is significant from a musicological and sociological standpoint. Influence relationships can help us better understand the historical development of genres and the overall evolution of music over time. For instance, why are certain musical elements more enduring than others and hence become more influential over time? Besides knowledge discovery however, inferring musical influence also has practical application. With today's vast quantity of available music metadata and music audio, which in and of itself is a form of data, there exists a need for new methods of cataloging and organizing it all. Influence relationships perhaps serve as one such means of making sense of this data.

The scale of data available today however has a silver lining though— while creating new challenges, it also presents an opportunity to study music influence in a data-driven way that was not possible until recently. With the vast availability of album metadata, cover song listings, collaboration information, lyrics and song audio available on the internet, there are many plausible approaches to tackling the problem of modeling musical influence. Though far from being exhaustive, we explore several of these approaches in this thesis.

1.2 Related Work

Previous work has been done on analyzing known sample-based musical influence networks [3], but with the exception of the work of Collins [5, 6], Shalit et al. [18], and Morton et al.[14], there has been limited work done on the task of inferring musical influence relationships through data.

Nick Collins, perhaps one of the earliest to research musical influence recognition, investigated content-based classification of Synth Pop tracks on a small manually annotated dataset of 364 tracks [5]. Later he experimented with Prediction by Partial Match (PPM) variable order Markov models, but again the dataset used was relatively small (248 tracks) [6].

Shalit et al. [18] presented the first study of musical influence at scale using a topic modeling approach. Specifically, they used the dynamic topic model [1] and document influence model [7], time series extensions to traditional topic modeling which allow for the evolution of topics over time.

With the recent surge in popularity of deep learning based methods, Morton and Kim presented the first application of deep learning to content-based musical influence recognition [14]. They used a deep belief network for feature extraction from a spectral representation of audio, though they treated influence identification as a multi-label classification problem with only 10 total classes (influencing artists).

1.3 Our Contribution

This thesis explores methods for inferring musical influence relationships through data, focusing primarily on *content-based* methods using song audio. Specifically, first we explore a topic modeling approach to artist-topic influence using the Document Influence Model (DIM), using a larger scale dataset than has been used previously and a bag-of-words feature extraction procedure. Secondly, we present the first (to the best of our knowledge) approach to predicting song-level influence utilizing siamese convolutional neural networks trained on mel-spectogram representations of song audio, achieving a validation set accuracy of 0.7. We also apply our trained siamese network in proposing an algorithm for the relative ranking of influencers for a given artist. For evaluation of our results, we used as ground truth a network graph of critic-determined influence relationships between musicians scraped from AllMusic.

1.4 Thesis Outline

In the second chapter of this thesis, we detail the various data sources used in this project as well as the methods used to collect that data. The third chapter describes the exploratory analysis conducted in order to investigate the respective feasibilities of both a *network-based* approach, using cover song data from SecondHandSongs, as well as a *content-based* approach, using audio files scraped from AllMusic.

In the fourth chapter, we describe the first content-based approach we tried to model influence, the Document Topic Model. In contrast to previous attempts [18] to model artist influence using the DIM, we use a larger audio dataset that we scraped from AllMusic.com and a different feature representation than that presented by Shalit et al. Due to the limitations of such an approach, in the fifth chapter we move on to a deep learning strategy using siamese convolutional neural networks for binary classification of artist-to-artist influence and discuss an application of such a strategy in the ranking of musical influencers.

Supporting code and data can be found at
https://github.com/xueharry/music_influence.

She a pretty penny and she know I'm doing numbers
Till we crash up the whole database

"Paradise"- Big Sean

2 Data

2.1 Sources

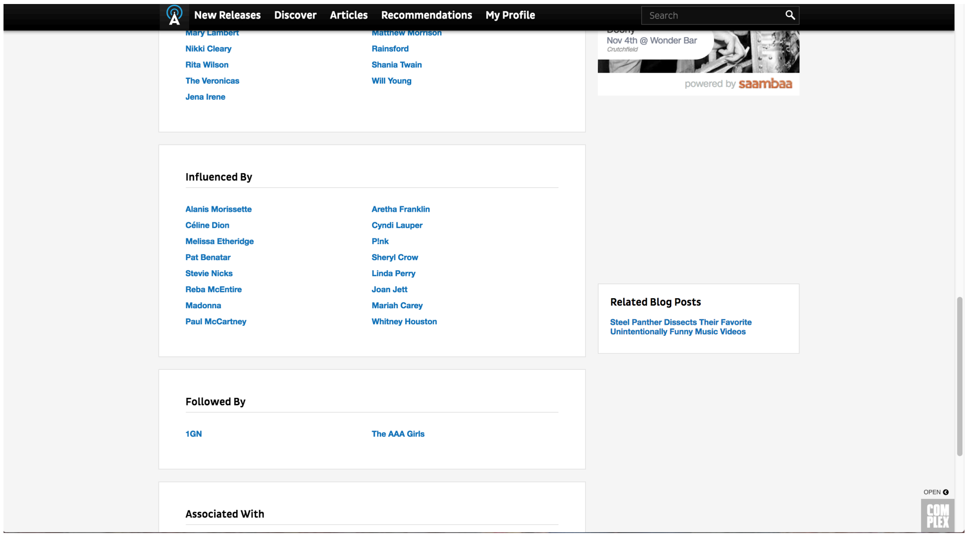
We used influence and audio data from AllMusic[1], cover song data from SecondHandSongs[2], and collaboration data from MusicBrainz[3]. Additionally, song release year information was queried for via Discogs[4].

2.1.1 AllMusic

True to its name, AllMusic is the largest music database on the web, cataloging information on over 3 million albums and 30 million tracks, along with associated artist information and other metadata.

[1] https://www.allmusic.com/
[2] https://secondhandsongs.com/
[3] https://musicbrainz.org/
[4] https://www.discogs.com/

AllMusic maintains individual Artist pages, which among other information includes human-curated data on a particular artist's *influencers* (denoted as "Influenced By") and *followers* (denoted as "Followed By"). The site defines influencers as "Artists that have had a direct musical influence on, or were an inspiration to, the selected artist, as determined by our music editors" and followers as "Artists who were influenced by the selected artist. This may be directly called from research and interviews, or it may be a strong inference based on the opinion of the editors" [5].



Mary Lambert
Nikki Cleary
Rita Wilson
The Veronicas
Jena Irene

Matthew Morrison
Rainsford
Shania Twain
Will Young

Influenced By

Alanis Morissette
Céline Dion
Melissa Etheridge
Pat Benatar
Stevie Nicks
Reba McEntire
Madonna
Paul McCartney

Aretha Franklin
Cyndi Lauper
P!nk
Sheryl Crow
Linda Perry
Joan Jett
Mariah Carey
Whitney Houston

Followed By

1GN

The AAA Girls

Associated With

Related Blog Posts

Steel Panther Dissects Their Favorite Unintentionally Funny Music Videos

Figure 2.1.1: Example of influencers and followers for an Artist on AllMusic

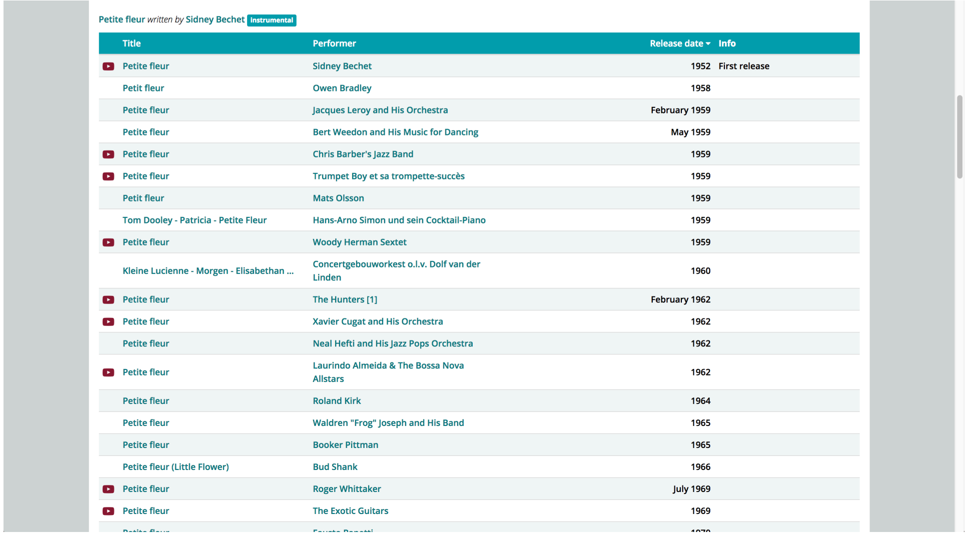
In addition to textual metadata, AllMusic also includes audio data in the form of a series of several (up to 10) 30 second long previews of songs recorded by a particular Artist.

2.1.2 SecondHandSongs

SecondHandSongs (SHS) is a cover songs database with an emphasis on data quality. For each original work, the site maintains information on the original

[5] https://www.allmusic.com/faq

recording and known subsequent versions recorded by other artists, commonly referred to as *covers*. Each cover has information on the performer that recorded the work along with the release date of the cover. Visitors to the site can post suggestions for new covers, but each version is verified by a human editor prior to inclusion in the database.



Petite fleur *written by* Sidney Bechet Instrumental

	Title	Performer	Release date ▾	Info
▶	Petite fleur	Sidney Bechet	1952	First release
	Petit fleur	Owen Bradley	1958	
	Petite fleur	Jacques Leroy and His Orchestra	February 1959	
	Petite fleur	Bert Weedon and His Music for Dancing	May 1959	
▶	Petite fleur	Chris Barber's Jazz Band	1959	
▶	Petite fleur	Trumpet Boy et sa trompette-succès	1959	
	Petit fleur	Mats Olsson	1959	
	Tom Dooley - Patricia - Petite Fleur	Hans-Arno Simon und sein Cocktail-Piano	1959	
▶	Petite fleur	Woody Herman Sextet	1959	
	Kleine Lucienne - Morgen - Elisabethan ...	Concertgebouworkest o.l.v. Dolf van der Linden	1960	
▶	Petite fleur	The Hunters [1]	February 1962	
▶	Petite fleur	Xavier Cugat and His Orchestra	1962	
	Petite fleur	Neal Hefti and His Jazz Pops Orchestra	1962	
▶	Petite fleur	Laurindo Almeida & The Bossa Nova Allstars	1962	
	Petite fleur	Roland Kirk	1964	
	Petite fleur	Waldren "Frog" Joseph and His Band	1965	
	Petite fleur	Booker Pittman	1965	
	Petite fleur (Little Flower)	Bud Shank	1966	
▶	Petite fleur	Roger Whittaker	July 1969	
▶	Petite fleur	The Exotic Guitars	1969	
	Petite fleur	Fausto Papetti	1970	

Figure 2.1.2: Example of cover versions for a song on SecondHandSongs

2.1.3 MusicBrainz

MusicBrainz is an open database of music metadata on artists and recordings. While the database is quite extensive, we used it for the limited purpose of extracting collaboration relationships between musicians. In this case, two artists are said to have collaborated if there exists a recording on which the artists are both listed within the MusicBrainz database.

2.1.4 Discogs

Discogs is a crowdsourced database about audio recordings, with information on over 9 million releases by over 5 million artists. Though accuracy is a concern because of the nature of crowdsourcing, due to its easy-to-use public API, we used Discogs in order to collect song release year information.

2.2 Collection

Scraping AllMusic

Since AllMusic does not have a free public API, we scraped influence information from the website directly. AllMusic provides a link to the respective Artist pages of the influencers and followers of a given Artist, which enabled us to construct a directed graph of influence relationships via breadth-first search (BFS).

We started on the Artist page of the jazz saxophonist Charlie Parker, adding a directed edge leading to Charlie Parker for each of his influencers and a directed edge leading away from Charlie Parker for each of his followers. We then added the associated Artist page URLs of each of the influencers and followers to a queue for exploration via BFS. When visiting each Artist page, we additionally collected metadata on the artist visible on the page, namely the active period of their career (i.e. 1930s - 1950s for Charlie Parker), and associated genres and styles.

A natural assumption of this approach is that starting the breadth-first search from Charlie Parker covers a sufficient number of genres and periods of influence relationships. This assumption is reasonable, given that our approach generated over 90,000 influence relationships between over 16,000 artists spanning genres including jazz, rock, country, classical, electronic, rap and pop and periods dating from the early 20th century to the present day. That said, a caveat is that since BFS from a single initial node necessarily yields a single weakly connected component, any influence relationships not connected to this component will be missing in our data.

We also collected audio data from AllMusic. Using the unique identifiers for each artist obtained during the scraping of influence relationships, we scraped AllMusic once again for available audio clips of songs by the artist. Since the number of audio clips we needed to scrape was much larger than the number of Artist pages visited during the initial BFS scraping process, we distributed audio scraping between multiple machines on Amazon Web Services (AWS).

Scraping SecondHandSongs

SecondHandSongs has a public facing API, but unfortunately the API does not return recording date information, which was essential for our purposes. Therefore we scraped the individual Work pages from the website in order to have access to the sequence of covering artists and cover release dates for each original musical work. Since each work page on SHS is indexed numerically by id (i.e. https://secondhandsongs.com/work/<id>), we queried all ids between 1 and 200000 (since not all ids are yet defined), distributing the scraping process between multiple machines on AWS.

MusicBrainz Collaboration Data

An undirected graph of collaboration relationships between musicians who made a recording together in the MusicBrainz database has previously been constructed[6], so we used it directly.

Querying the Discogs API for Release Year Information

Discogs exposes a public API for requesting song metadata from its database. Using this API we were able to fuzzy search for song release year based on artist name and song name, as there was no way to access this information from AllMusic.com directly for the audio that we scraped. In total, we were able to collect release year information for 126024 songs out of 138008 total, obtaining 91% coverage.

[6] https://github.com/basimr/snoop-dogg-number/tree/master/graph

Let's keep the night fantastic
Light it up, tell me more, explore

"Who Do We Think We Are?"- John Legend

3
Exploratory Analysis

Before performing any modeling, we first examined the feasibility of two potential approaches to our problem in terms of the type of data used, a *network-based* approach, using cover song data vs. a *content-based* approach, using song audio directly.

3.1 Analysis of Network Data

3.1.1 AllMusic Influence Network

The AllMusic influence network is a sparse graph consisting of 16,704 artists and 93,065 influence relationships with each artist having an average of 5.57 followers. Both the indegree and degree distributions are heavily right skewed, which makes intuitive sense as most artists have relatively few followers, while extremely influential artists have many followers.

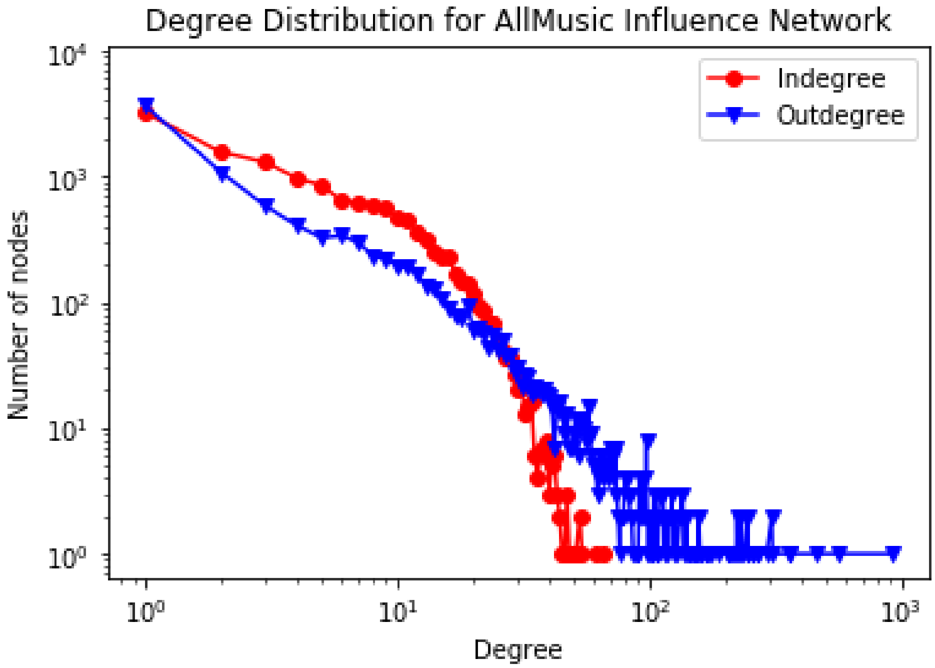


Figure 3.1.1: Degree distributions for AllMusic influence network

Highest Out-Degree Artists

Ordered by degree, the top 25 artists that directly influenced the highest number of followers are listed in Table 3.1.1, followed by the count of artists they influenced.

Table 3.1.1: Artists with highest out-degree in AllMusic influence network

Artist	out-degree
The Beatles	911
Bob Dylan	558
The Rolling Stones	463
David Bowie	358
The Velvet Underground	356
Jimi Hendrix	308
The Beach Boys	306
The Kinks	306
Led Zeppelin	291
Neil Young	269
Miles Davis	266
James Brown	260
The Byrds	259
Black Sabbath	245
John Coltrane	244
Hank Williams	243
The Stooges	241
Brian Eno	237
The Who	230
Pink Floyd	227
Ramones	225
The Clash	224
Kraftwerk	222
Elvis Presley	222
Sex Pistols	220

The genre of rock has the highest representation in this list, with artists from jazz, electronic and pop appearing as well.

PageRank

Since outdegree only takes into account first-order relationships in a graph, we also computed PageRank [16] over the AllMusic influence graph. Edge directionality in the graph was reversed before applying PageRank (yielding follower to influencer directed edges), since the "authorities" in this specific context are the influencers. Originally devised to rank website importance, PageRank computes the stationary distribution of a random walk over a network graph. The resulting PageRank vector is a vector of probabilities which sums to 1, with the corresponding PageRank value for each node indicating the proportion of time expected to be spent at that node during such a random walk. The top 25 artists in terms of PageRank are summarized in the table below:

Table 3.1.2: Artists with highest PageRank in AllMusic influence network

Artist	PageRank
Louis Armstrong	0.00723579
Scott Joplin	0.00692252
The Beatles	0.00642019
Charley Patton	0.00484325
Jelly Roll Morton	0.00465115
Uncle Dave Macon	0.00447063
Fats Waller	0.00427571
Bob Dylan	0.00374972
Jimmie Rodgers	0.00357579
James Brown	0.00350958
King Oliver	0.00336848
James P. Johnson	0.0031979
Duke Ellington	0.00317865
Chuck Berry	0.00305074
Louis Jordan	0.00303284
W.C. Handy	0.00298179
Mike Walbridge	0.00292283
The Rolling Stones	0.00287099
Blind Lemon Jefferson	0.002761
The Mills Brothers	0.00270951
The Velvet Underground	0.00265675
Bessie Smith	0.00249454
Little Richard	0.00246647
Hobart Smith	0.00245348
Jimi Hendrix	0.00242845

Though there is overlap between the respective top 25 artists for out-degree and PageRank (The Beatles and The Rolling Stones for instance), as expected PageRank does uncover artists who do not necessarily have high outdegree but are nevertheless authoritative in terms of being influences of artists who themselves are influential. For example, the table above includes pivotal figures in 20th century music such as Louis Armstrong, Charlie Patton, and Chuck Berry, who would be missed by a simple out-degree analysis.

Breakdown by Genre

Table 3.1.3: Proportion of artists belonging to each genre

Genre	Proportion
Pop/Rock	0.430136
Jazz	0.087524
R&B;	0.065852
Unknown	0.065254
Rap	0.057890
Electronic	0.057771
Country	0.044959
Latin	0.025682
Blues	0.024545
International	0.020175
Vocal	0.018738
Folk	0.018259
Religious	0.016403
Reggae	0.015805
Classical	0.015386
Comedy/Spoken	0.010836
Avant-Garde	0.007663
New Age	0.006765
Stage & Screen	0.006645
Easy Listening	0.002814
Children's	0.000838
Holiday	0.000060

Unsurprisingly, we see that Pop/Rock is overrepresented in this data set, followed by Jazz, R&B, Rap, Electronic and Country. 6.5% of artists do not have genre labels associated with them.

Influence Between Genres

To visualize the amount of influence between genres as represented by the AllMusic influence graph, we created the heatmap in the figure below:

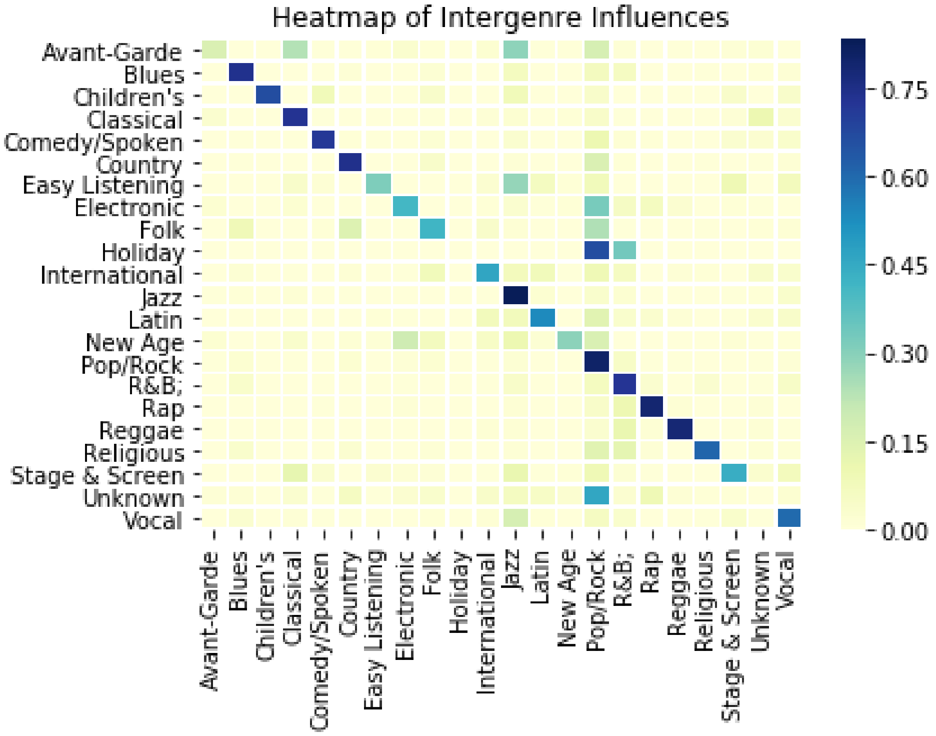


Figure 3.1.2: Heatmap of intergenre influence in AllMusic influence graph

In order to construct the heatmap, we used the genre metadata we scraped for each artist, using the first genre tag for artists with multiple genre tags to calculate the frequencies of edges from each genre to every other genre, normalizing by the total number of edges originating from each genre. Therefore the heatmap can be read as follows: row *M* column *N* designates the proportion of influence genre *M* contributes to genre *N* (darker hue meaning higher contribution) as suggested by the network graph, with the proportions in each row summing to 1 and the diagonal entries indicating how self-contained or "insular" a genre is.

We observe that:

- A majority of genres give their second highest influence contribution (outside of to themselves) to Pop/Rock, which is not surprising given the conglomerate nature of Pop/Rock as a genre
- Avant-Garde is relatively evenly spread in its influence between itself, Classical, Jazz and Pop/Rock
- Jazz, Pop/Rock and Rap appear to be the most "insular" genres
- Blues influences Jazz, Pop/Rock and R&B, which is consistent with conventional wisdom

It is important to note the limitations of using genre information from AllMusic. First, we observe that Pop/Rock are lumped into one category, which is not ideal as two discrete categories for the two genres would be more informative. Secondly, since AllMusic reflects popular music tastes across the past century, we see over-representation of artists from genres such as Jazz and Pop/Rock in the influence graph, which skews results in the heatmap as well.

3.1.2 SecondHandSongs Covers

After dropping covers with missing performer and release date information from the SHS cover data, we were left with 644,786 total versions (covers) of 86,827 unique works from 77,328 unique artists.

Distribution of Number of Covers per Work

Grouping versions together by the original work they are associated with, we calculated basic summary statistics:

Table 3.1.4: Summary statistics for number of covers per original work from SecondHandSongs

count	86827.00000
mean	7.42610
std	25.23396
min	1.00000
25%	2.00000
50%	2.00000
75%	5.00000
max	2004.00000

This distribution is also heavily right-skewed, with a median count of 2 covers per original work and a mean of 7.426.

Most Covered Works

We extracted the top 25 most covered works from SecondHandSongs, along with counts of the number of times they were covered.

Table 3.1.5: Most covered works from SecondHandSongs

Work Name	Covers
Silent Night! Holy Night!	2004
Summertime	1611
Away in a Manger [Mueller]	1536
O, Holy Night	1304
New Britain	858
White Christmas	828
Have Yourself a Merry Little Christmas	825
O Come, All Ye Faithful	808
Can't Help Falling in Love	804
The Christmas Song (Merry Christmas to You)	761
Over the Rainbow	709
Body and Soul	707
What Child Is This?	664
Winter Wonderland	615
God Rest You Merry, Gentlemen	612
Jingle Bells	608
The First Nowell the Angel Did Say	605
My Funny Valentine	579
Stille Nacht! Heilige Nacht!	545
Yesterday	538
I'll Be Home for Christmas (If Only in My Dreams)	538
Carol of the Drum	534
Joy to the World	531
St. Louis Blues	521
Love Me Tender	520

We see highest representation from Christmas songs and jazz standards in this list.

3.1.3 MusicBrainz Collaboration Network

The MusicBrainz collaboration network is an undirected graph consisting of 271,442 nodes (artists) and 650,920 edges with average degree 4.796. The graph is not connected and instead consists of 26,654 separate connected components.

3.1.4 Overlap Analysis Between Datasets

Overlap Between Influence Network and Cover Songs

We first calculated the node overlap between the AllMusic influence network and artist names in the SecondHandSongs dataset using exact string matching. The node overlap found using this method was 57.24%, which perhaps reflects the limitations of this approach.

We also calculated edge overlap between the influence network and the cover songs data. Obviously, the cover song data does not form a network on its own. In fact, one natural way of viewing the sequence of covers for a given original work is that each cover sequence is an observed *trace* of information diffusion across a latent directed network of influence between musicians.

Therefore we used three different underlying assumptions for network formation in order to establish a baseline for edge overlap between the influence and cover song data:

1. *Next immediate chronological neighbor*: creating directed edges between each artist and the next immediate artist chronologically that covered the same original work

2. *First artist to each successor*: creating directed edges between the first artist that covered an original work and each of the subsequent artists who covered the original work

3. *Each artist to every possible successor*: creating directed edges between each artist and every subsequent artist in the cover sequence for the song

The edge overlaps between AllMusic and SHS for each of these edge creation assumptions are summarized in the table below:

Table 3.1.6: Edge overlap based on cover song edge creation assumption

Assumption	Number of Edges Overlap	Percentage Overlap
1	2951	3.55%
2	6461	7.77%
3	14668	17.6%

Assumption (3) yielded the highest overlap, which is not surprising given that it generates the highest number of possible "edges". Allowing for duplicates (2 artists who were in the same cover sequence for multiple songs), assumption (3) yielded 82,558 "edges" that were found in the ground truth, which means that artists will often cover their influencers' original works more than just once.

Overall, the percentage overlap is not very high for any of the methods. One possible reason is the imperfect approach of using exact string matching on artist names between the two datasets, so there may be discrepancies created by handling of special characters, alternate spellings, variations of artist names etc.

Overlap Between Influence Network and Collaboration Network

We also calculated the node and edge overlap between the AllMusic influence network and the MusicBrainz collaboration network. Again, we used exact string matching on artist name between the two datasets.

The node (artist name) overlap between the two datasets is 63.69%, which is comparable to the artist name overlap between AllMusic and SHS. By overlap, we refer to the number of artist names in the smaller influence network that are found in the collaboration network divided by the total number of nodes in the influence network.

We also calculated edge overlap between the two datasets. Since the collaboration network is undirected, for each undirected edge (u, v) in the graph, we introduced two directed edges (u, v) and (v, u) for the purpose of calculating overlap. The calculated overlap is 3.53%, which is far lower than the overlap between the influence network and cover songs data. This simple heuristic suggests that collaboration relationships are not especially directly predictive of influence relationships, which is reasonable given that many musicians never get the opportunity to record with their influences.

3.1.5 Limitations of Network-based Approach

Regardless of the method used to construct a network out of cover song data, whether one of the heuristics mentioned in 3.1.4 or a cascade-based influence algorithm such as in [8], the use of cover song data arguably poses two fundamental challenges in influence inference, which we will refer to as the **standards effect** and the **career cover artist effect**.

By the standards effect, we mean that certain songs are covered very often simply because they are a common part of the repertoire (think Christmas songs or jazz standards). This behavior obscures true influence relationships and is commonly referred to in network terminology as **herding**. Evidence for this phenomenon can be seen in Table 3.1.5, where we see that many of the most covered works are precisely Christmas songs or jazz standards. By the career cover artist effect, we refer to the fact that certain artists almost exclusively record cover songs, where the covers are often recorded for commercial reasons or other non-influence reasons.

These issues could be dealt with through certain heuristics, for example removing songs that have over a certain number of covers. However, combined with the poor node overlap issue, perhaps this suggests that a network-based approach using cover songs is not the best way to address our task due to the underlying signal being too weak.

3.2 Analysis of Audio Data

3.2.1 Coverage

Since we scraped the audio data directly from the AllMusic website matching on the unique Artist identifier for the site, we did not run into the coverage overlap issues that we did with the cover song or collaboration datasets. We were able to collect audio clips for 92.55% of artists in the AllMusic influence network, which accounts for 95.64% of the influence edges in the ground truth influence dataset.

We were able to extract a mean of 9.29 30-second long clips per artist, with less than 14% of artists having fewer than 10 audio clips and less than 7% having fewer than 5 clips. In total, 143,625 clips of audio were collected.

3.2.2 Intermediate Feature Representation

For our exploratory analysis, the raw audio files were far too large to use directly and high-level engineered features such as those used in [18] can be overly lossy. Therefore we struck a balance between these extremes through the use of 2 types of intermediate time-frequency feature representations commonly [20] used in audio signal processing, **mel-spectrograms** and **mel-frequency cepstral coefficients** (**MFCCs**). The **mel scale** performs a logarithmic transformation of frequencies to more closely approximate the way humans perceive pitch distances.

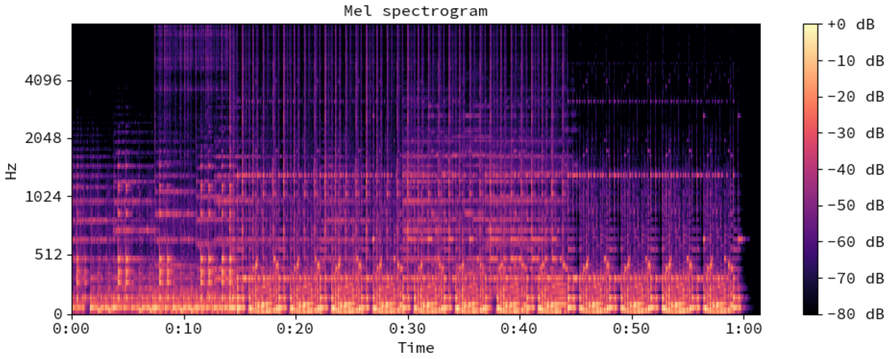


Figure 3.2.1: Example of mel-spectrogram representation of audio file

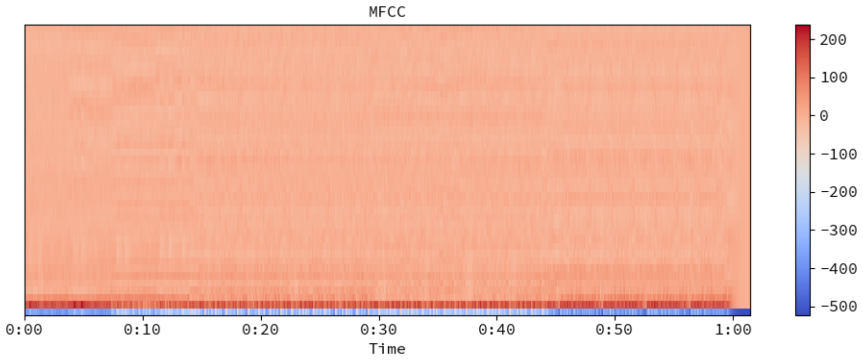


Figure 3.2.2: Example of MFCC representation of audio file

Practically, both the mel-spectrogram and MFCC representation of a given song are 2-dimensional arrays of floating-point numbers with the first dimension corresponding to the frequency domain and the second dimension corresponding to the time domain.

3.2.3 Dimensionality Reduction with PCA

We used Principal Component Analysis (PCA) for dimensionality reduction. PCA takes a set of data and transforms it into a new orthonormal coordinate system where the first coordinate (first principal component) explains the most variance, the second principal component explains the second most variance and so on.

Taking the MFCC features corresponding to the first AllMusic audio sample for each artist, we extracted the first 2 principal components, which together explained approximately 55 percent of the variance in the data. We then visualized the projection of the MFCC features onto the first two principal components, colored by genre and with text labels for the top 3 highest out-degree artists per genre. For increased readability, we only included data points from the 7 most popular genres in terms of total number of artists. The visualization can be seen in Figure 3.2.3.

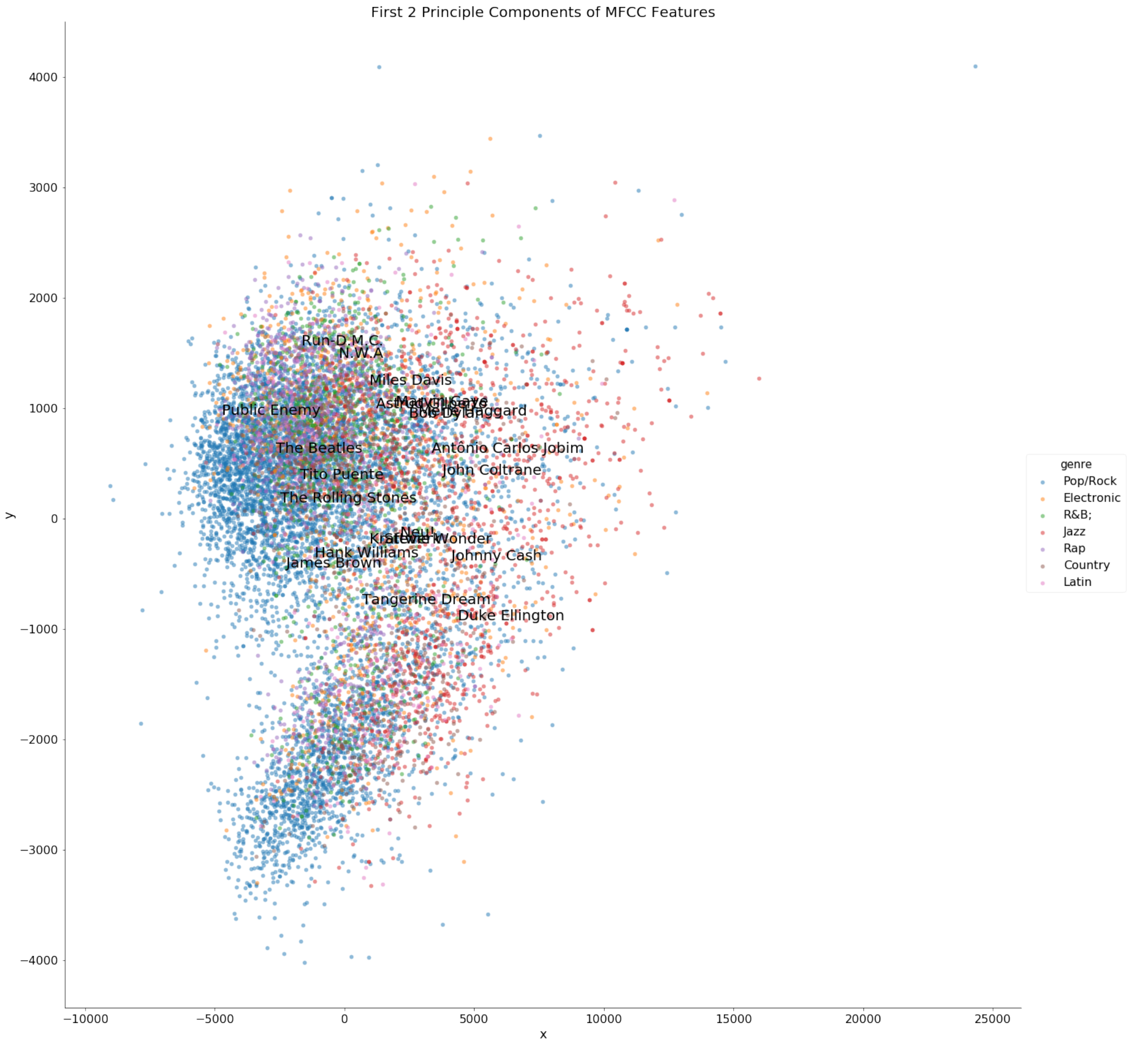


Figure 3.2.3: PCA projection of MFCC features

Note that we only used one 30 second clip to represent each artist, but even so there are readily discernible patterns. The highest degree Rap artists, N.W.A., Run-D.M.C. and Public Enemy are all localized to the top left of the plot, where there appears to be a large cluster of other Rap artists. The Jazz genre seems to be predominantly located in the right half of the plot while the bulk of R&B is located between Rap and Jazz, which is consistent with both the chronology and stylistic progression relationship between these three genres.

3.2.4 Relationship Between Influence Graph Distance & Euclidean Distance on PC Projection

We also investigated the relationship between node distance in the AllMusic Influence Graph and Euclidean distance in the principal component projection. To do this we computed the average Euclidean distance in the principal component projection between each artist and all descendants at breadth-first search (BFS) distance exactly *d* followers away in the influence graph for increasing values of *d*.

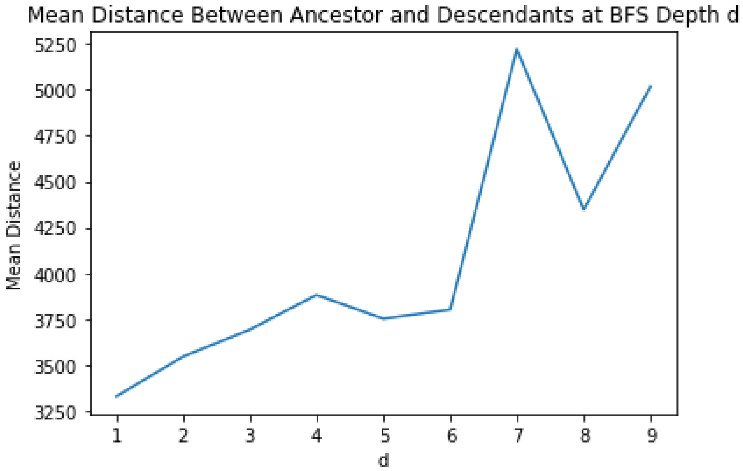


Figure 3.2.4: Average Euclidean distance between nodes in PC plot vs. BFS depth in influence graph

We see that mean Euclidean distance roughly increases with increasing BFS depth, which provides evidence that the PCA projection structure approximately corresponds to the influence network in terms of influencer-follower distance.

Overall, from our exploratory analysis we saw that there were many limitations to using a network-based approach and that the content-based approach appeared more promising. Therefore, we decided to focus on content-based approaches, which constitute the remainder of this thesis.

All our history hidden, ain't no liberty given
We all fit the description of what the documents written

"Land of the Free"- Joey Bada$$

4 Inferring Artist Influence with the Document Influence Model

As a first content-based approach to inferring artist influence based on audio samples, we used the Document Influence Model (DIM) [7] developed by Gerrish and Blei.

The Document Influence Model is an extension to traditional topic modeling which allows for the evolution of topics over time. Though originally developed for text documents, the DIM also makes sense in the context of music since music consists of multiple genres and subgenres which also mix and evolve over time. Furthermore, since this model has previously been applied in a similar way by Shalit et al. [18] we used it as a baseline check for our data pipeline.

4.1 Model

The DIM is a probabilistic time series model with the following three components:

1. Latent Dirichlet Allocation (LDA) [2] fit separately on each time epoch, with each epoch corresponding to a year of song release in this case.

2. Time evolution: Each topic evolves with time, linking the different epochs.

3. Song-topic influence factor: Each song has a hidden associated influence factor for each topic whose value is revealed via posterior influence. Therefore, in this model influential songs are defined as songs that "pull" the language of later songs in their topic in their direction.

Formally, we have a corpus of D songs (documents) where each song $d \in \{1...D\}$ consists of a set of N_d musical words $w_1^d, ..., w_{N_d}^d$ drawn from a vocabulary of total size W. Each song belongs to one of T time epochs (song release year, though we also experimented with using the start year of the artist's career), and we assume K total topics.

Each word w_n^d is generated from one topic $k \in \{1...K\}$, with topic assignment indicated by the variable $z_{n,k}^d$. Since each song is a bag-of-words representation over the topics, then therefore $\frac{1}{N_d}\sum_{n=1}^{N_d} z_{n,k}^d$ represents the proportion of each topic k in song d.

The probabilistic model used is defined as follows: The word distribution at time t for topic k is given by a W-dimensional natural parameter vector $\beta_{k,t}$, with the probability of a word w given by the softmax transformation:

$$p(w|\beta_{k,t}(w)) \propto exp(\beta_{k,t}(w))$$

The topic-term distribution drifts over time via the stationary autoregressive process

$$\beta_{k,t+1}|\beta_{k,t} \sim \mathcal{N}(\mu_{k,t}, \sigma^2 I)$$

where σ^2 is the transition variance and

$$\mu_{k,t} = \beta_{k,t} + exp(-\beta_{k,t}) \sum_d \ell_k^d \cdot \kappa(t, \tau(d)) \sum_n w_n^d z_{n,k}^d$$

where the first component of the sum is the topic-term distribution in the previous time-epoch and the second component of the sum is the sum of the songs in the previous epochs, scaled by their influence score and a time-delay kernel (in this case, a log-normal kernel was used), denoted by $\kappa(t, \tau(d))$ with $\tau(d)$ representing the release year of the song. w_n^d is an indicator defined to be 1 if the nth word w_n appears in document d and 0 otherwise. Each song is given a normally distributed topic-influence score ℓ_k^d which denotes how much the language of topic k drifts in the direction of the language of song d.

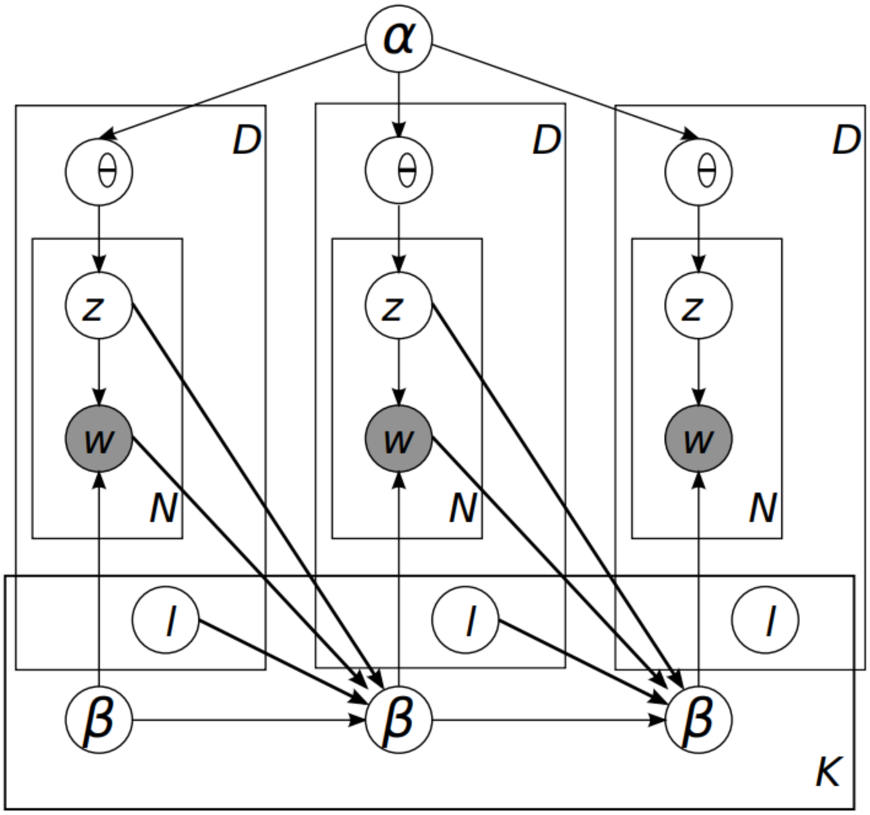


Figure 4.1.1: Plate diagram of the Document Influence Model

As the exact posterior distribution is intractable, Gerrish & Blei derived a variational approximation using Kalman filters [7], the details of which are omitted here.

Posterior inference enables us to estimate the topic-influence scores ℓ_k^d (Note: ℓ is written as l in the plate diagram in figure 4.1.1), which is the key variable of interest. We defined the influence of each song as $\ell^d = \max_k \ell_k^d$, and for each artist $a \in \{1...A\}$, we set the influence for the artist ℓ^a as the average over all ℓ^d corresponding to songs by that artist.

4.2 Experimental Setup

4.2.1 Feature Representation

As is generally the case with topic modeling, the DIM also requires that each song (document) be represented as a bag-of-words (BOW). Previously, Shalit et al. [18] used features from the publicly available Million Songs Dataset, and also engineered music domain specific features such as max. loudness, chroma and timbre.

Since we did not use the Million Songs Dataset, instead relying upon raw audio scraped directly from AllMusic to maximize overlap with the ground truth influence graph, we had to generate audio features ourselves. To this end, we used a common procedure [13] for generating a bag-of-words representation for the MFCC representation of each audio track. For reference, the MFCC representation for each audio file was a (number_of_features, number_of_frames) array of floats with each frame corresponding essentially to a timestep. In our case, we had a $(13, 1298)$ array for each MFCC representation, corresponding to the first 13 MFCC coefficients over 1298 frames (approximately 30 seconds). The bag-of-words generation procedure used is as follows:

1. Normalize each MFCC coefficient by subtracting the mean and dividing by the standard deviation across the entire audio dataset for that coefficient.

2. Cluster all normalized 13-dimensional frames across the entire audio dataset using minibatch *k*-means [17]. *k* corresponds to the desired dimensionality (vocabulary size) of the end bag-of-words representation.

3. For each normalized MFCC representation, quantize each frame by assigning the frame to the nearest cluster center, tallying the counts of assignments for each cluster over all frames to obtain a bag-of-words.

4.2.2 Model Fitting

Since inference for the Document Influence Model with the scale of data that we used was too RAM-intensive, we performed model fitting on Harvard's Odyssey cluster. Specifically, we fit the DIM on 125,965 total songs compared to the 24,941 songs used by Shalit et al. The breakdown of number of songs for each epoch (year of release) can be seen in the figure below.

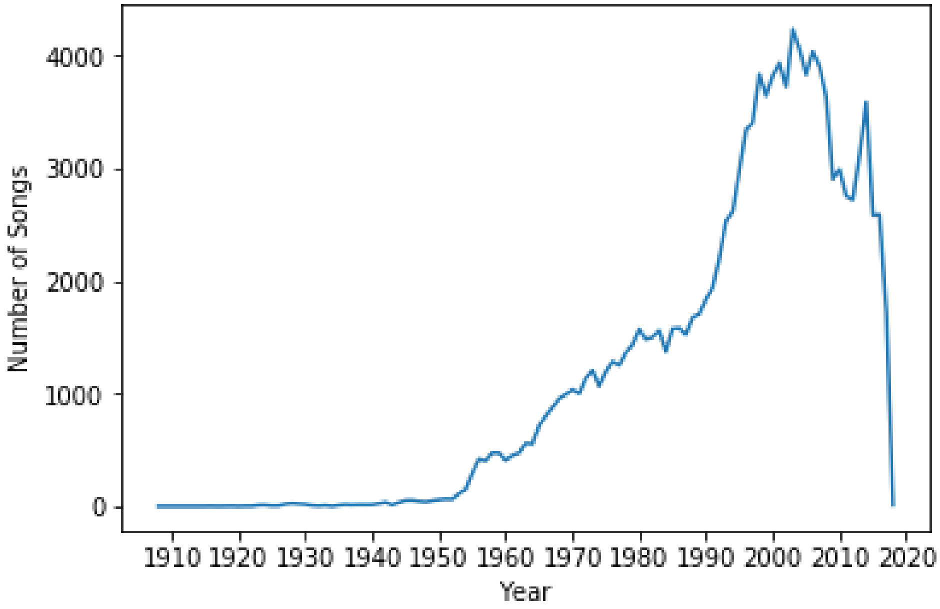


Figure 4.2.1: Number of songs per year used to fit DIM

In our experiments, we tried out bag-of-words sizes of 500 and 1000 and number of topic settings 1, 5 and 10. Due to the amount of time it took to generate features according to the procedure described in 4.2.1, we did not optimize for the selection of *k* in *k*-means for our bag-of-words size, though strategies such as the elbow method or the gap statistic [19] certainly could have been used.

4.3 Results

4.3.1 Correlation with AllMusic Influence Graph

To evaluate the model, we calculated the Spearman correlation coefficient between artist influence score according to the unsupervised model and artist out-degree from the ground-truth influence graph that we scraped. The results for several configurations are summarized in the table below (all statistically significant with $p < 0.05$):

Table 4.3.1: Correlation of DIM Influence with AllMusic Outdegree

BOW Size	Number of Topics	Correlation
500	1	0.1303
500	5	0.1687
500	10	0.1733
1000	1	0.1052
1000	5	0.1819
1000	10	0.1691

4.3.2 Most Influential Artists from Best DIM Model

We computed the top 25 most influential artists according to the highest-correlated model (BOW Size 1000, 5 topics) in terms of ℓ^a (as defined in section 4.1), filtering for artists with out-degree greater than 100 in the ground truth graph. We applied this filtering in order to counteract noise in the model fitting process from some time epochs having very few total songs, which led to some artists with very low out-degree having an inflated value of ℓ^a. To reemphasize however, the DIM is an unsupervised model with no out-degree information or other metadata used during the fitting process, just the BOW feature representations of song audio.

Table 4.3.2: Most influential artists according to best DIM Model

Rank	Artist
1	Bob Marley
2	Parliament
3	Stevie Wonder
4	Frank Zappa
5	Prince
6	Louis Armstrong
7	The Band
8	Curtis Mayfield
9	The Clash
10	Ray Charles
11	Kiss
12	Funkadelic
13	The Yardbirds
14	Tom Waits
15	The Who
16	Otis Redding
17	Elvis Costello
18	MC5
19	T. Rex
20	Kraftwerk
21	New Order
22	James Brown
23	Alice Cooper
24	Buddy Holly
25	Ella Fitzgerald

The artists in the table above do not necessarily have the highest out-degrees (compare with table 3.1.1). We see that like PageRank (table 3.1.2), the DIM identifies Louis Armstrong as a highly influential artist despite his out-degree not being in the top 25. Overall, qualitatively this list looks reasonable: it includes pioneers of jazz (Louis Armstrong, Ella Fitzgerald), rock (Buddy Holly, The Yardbirds), electronic (Kraftwerk), funk (Parliament and its sister act Funkadelic), soul (Otis Redding, Curtis Mayfield, James Brown, Ray Charles) and punk (MC5). Obviously, these results are not without caveats: the filtering by out-degree step above does introduce bias and the relative rankings of artists is by no means definitive (Is Bob Marley *really* the most influential artist of all time, across all genres?).

4.4 Discussion

4.4.1 Comparison to Shalit et al.

Fitting the Document Influence Model on the audio dataset we gathered and through the bag of words feature extraction procedure described above, we achieve comparable results to what Shalit et al. obtained. For reference, Shalit et al. used audio features from the Million Songs Dataset in addition to additional engineered features to yield a larger bag-of-words vocabulary size of 5033, a smaller audio dataset of around 25k songs, and 12 total time epochs, achieving a top Spearman rank correlation of 0.15 using a 10 topic model.

In contrast, our best correlation was achieved with a 5 topic model with a bag of words size of 1000 and features generated by ourselves using the procedure described in section 4.2.1, with the slight boost in correlation probably attributable to the increase in amount of data.

4.4.2 Limitations

Though it yielded respectable results, a key limitation of the bag-of-words feature extraction we used is that much information is lost in the various stages of the feature extraction pipeline. First, in order to make clustering computationally tractable, we employ a MFCC feature representation, which is lossier than a mel-spectrogram representation. Next, we apply minibatch *k*-means, which produces lower-quality clusterings than standard *k*-means, which itself is a heuristic that is not guaranteed to find a globally optimal clustering in terms of minimizing loss (in fact, the *k*-means clustering problem in general is NP-hard). Finally, our resulting bag-of-words representation does not capture the rich temporal structure and substructure of music since it is merely a count summary that does not take order into account. Another limitation is that the DIM yields an influence score for each document on a per-topic basis, as opposed to the artist-to-artist level. With these issues of (1) a need for richer feature representations and (2) artist-to-artist influence modeling capacity in mind, we turn to exploring a deep learning approach to modeling artist-to-artist influence in the next chapter.

And your heart hurts, mine does too
And it's just words and they cut deep

"Look What You've Done"- Drake

5

Predicting Artist Influence with Siamese Networks

Due to the limitations of the Document Influence Model, the need for richer feature representations and the scale of the audio dataset we collected, we next decided to investigate the application of deep learning, specifically an architecture known as a siamese convolutional neural network in modeling song-level influence.

We address the following binary classification task: given as input a pair of two songs, what is the probability that there exists an influence relationship between the two artists who respectively recorded the songs?

5.1 Model

5.1.1 Overview

For the siamese network, we adopt a similar architecture as used by Koch et al. [12] in predicting image similarity. Originally used for predicting similarity of two input images where the total number of image classes is large and the number of training instances for each class is sparse, siamese networks have been shown to be very successful in learning powerful descriptive features which generalize well to unseen pairs.

At a high level, our model takes in as input two mel-spectrogram representations of songs, which can essentially be viewed as "images" of the songs. Specifically, each mel-spectrogram is a two-dimensional array of floating point numbers with the first dimension corresponding to the frequency domain (128 total frequency bins in our case) and the second dimension corresponding to the number of timesteps in the time domain. Each of these inputs is fed separately through two copies of the same convolutional neural network (CNN). These two "twin" (hence the name siamese) CNNs extract a high-level fixed-length vector representation for each of the songs. The component-wise absolute distances between the extracted vectors for each of the songs are calculated, fed through a fully-connected perceptron layer, and then a sigmoid activation function is applied to output a probability between 0 and 1. A diagram of the model we used, including the input and output shapes for each layer, can be seen in the figure below:

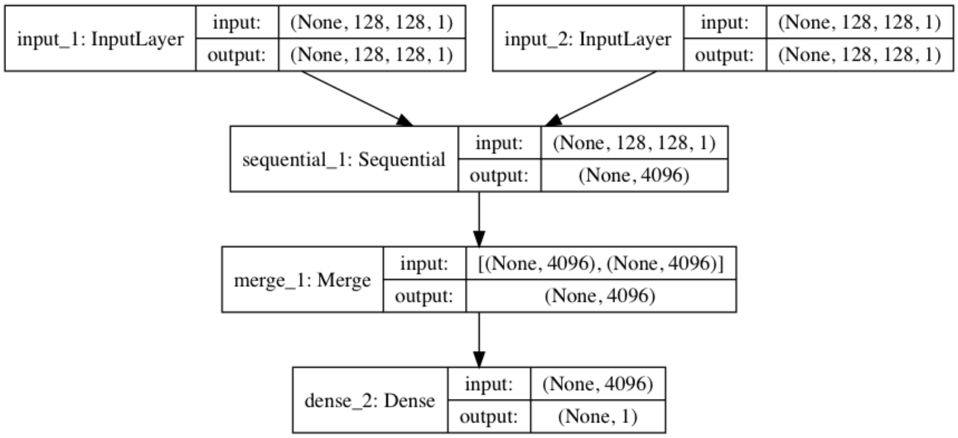

input_1: InputLayer
input: (None, 128, 128, 1)
output: (None, 128, 128, 1)
input_2: InputLayer
input: (None, 128, 128, 1)
output: (None, 128, 128, 1)
sequential_1: Sequential
input: (None, 128, 128, 1)
output: (None, 4096)
merge_1: Merge
input: [(None, 4096), (None, 4096)]
output: (None, 4096)
dense_2: Dense
input: (None, 4096)
output: (None, 1)

Figure 5.1.1: Diagram of siamese network architecture

`input_1` and `input_2` correspond to the 2 mel-spectrogram snippets of an input pair that we take in. A note on dimensions, taking the input dimensions of `input_1`, $(None, 128, 128, 1)$ as an example: The first dimension corresponds to batch size, which was left as *None* in the diagram for the sake of generality. The second and third dimensions correspond to the fact that each mel-spectrogram snippet used is a 128x128 matrix of real-valued numbers. The fourth dimension corresponds to the number of channels in the image, which was simply 1 in our case (in color image applications for instance, channel size is commonly set to 3 to deal with the RGB channels separately).

Each of the inputs are separately fed into the same CNN, called `sequential_1` in our diagram for feature extraction. Shortly, we will discuss this CNN in further detail.

`sequential_1` extracts a fixed-length vector representation of each song. In `merge_1` the absolute element-wise differences between the vector representations for each song are calculated, and then in `dense_2` the output is passed through one last fully connected layer with learnable weights, and a sigmoid activation function is applied.

The parameters for each of the layers of our siamese network are trained using the standard backpropagation algorithm against the standard binary cross-entropy loss function.

We implemented the model using the Keras library[4] in Python.

5.1.2 CNN Architecture

We now give a more detailed description of the CNN architecture we used (`sequential_1` in the figure above), depicted in detail in the figure below:

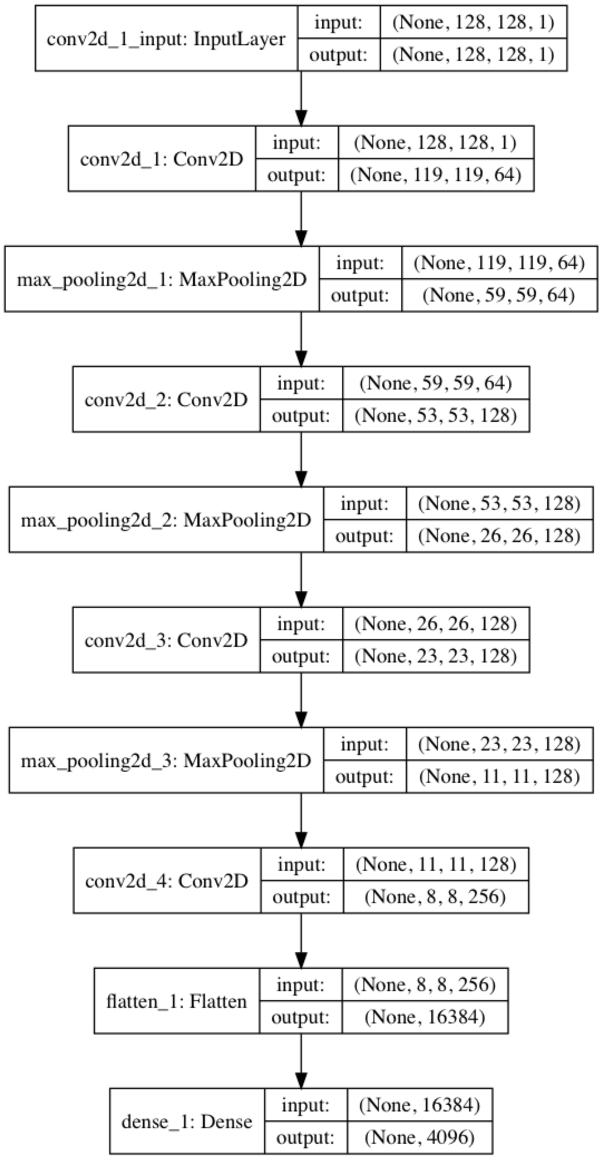

conv2d_1_input: InputLayer
input: (None, 128, 128, 1)
output: (None, 128, 128, 1)
conv2d_1: Conv2D
input: (None, 128, 128, 1)
output: (None, 119, 119, 64)
max_pooling2d_1: MaxPooling2D
input: (None, 119, 119, 64)
output: (None, 59, 59, 64)
conv2d_2: Conv2D
input: (None, 59, 59, 64)
output: (None, 53, 53, 128)
max_pooling2d_2: MaxPooling2D
input: (None, 53, 53, 128)
output: (None, 26, 26, 128)
conv2d_3: Conv2D
input: (None, 26, 26, 128)
output: (None, 23, 23, 128)
max_pooling2d_3: MaxPooling2D
input: (None, 23, 23, 128)
output: (None, 11, 11, 128)
conv2d_4: Conv2D
input: (None, 11, 11, 128)
output: (None, 8, 8, 256)
flatten_1: Flatten
input: (None, 8, 8, 256)
output: (None, 16384)
dense_1: Dense
input: (None, 16384)
output: (None, 4096)

Figure 5.1.2: Diagram of CNN architecture

Our CNN architecture consists of 4 convolutional layers with the ReLU activation function, alternated with 3 max pooling layers. The first convolutional layer consists of 64 10x10 convolutional filters, the second convolutional layer consists of 128 7x7 filters, the third convolutional layer consists of 128 4x4 filters, and the fourth convolutional layer consists of 256 4x4 filters. Max pooling is performed in between convolutional layers in order to downsample the image. Finally the results from the last convolutional layer are flattened and passed through a fully-connected layer with the sigmoid activation function before being passed to `merge_1` for calculation of distance between the feature vector representations for each song.

5.2 Experimental Setup

5.2.1 Sampling of Songs

Due to RAM limitations with the GPU we had access to for training and the need for reasonable training time, we were not able to use the entire mel-spectrogram representation corresponding to the full 30 seconds of audio we had access to for each song. Instead during training we randomly sampled contiguous 3 second samples on the fly from the full mel-spectrogram during each epoch (thereby generating different 3 second samples for each particular song each new epoch). Though using the full mel-spectrogram would have preserved the most information, given that previously [20] 3 second samples have been used in other music audio tasks involving deep learning and our eventual results, this was likely a reasonable simplification.

In addition to sampling at the song-level, to further simplify training time, we only used the audio from one song for each artist.

5.2.2 Train-Validation Split

As with any binary classification task, balance between the positive and negative example classes was a concern. In this case, our positive examples were derived from the ground truth AllMusic influence graph. Specifically, out of all edges in the ground truth graph, we had audio corresponding to 88,853 of them, and we used these pairs as our positive examples. We artificially generated negative examples of pairs where influence does not exist by randomly sampling for

88,853 artist pairs that do not correspond to edges in the ground truth graph, thereby creating a balanced number of positive and negative pairs overall. These combined 177,706 pairs were then randomly split into 80% training and 20% validation data.

5.2.3 Training

Training was performed on a Tesla K20Xm GPU on Harvard's Odyssey cluster. We used a batch size of 16 and the Adam optimizer [11] (an adaptive variant of standard Stochastic Gradient Descent), with training concluding after validation loss stopped decreasing for 5 epochs. In total, training took 51 epochs.

5.3 Results

The loss and accuracy curves on both the training and validation sets can be seen in the figures below:

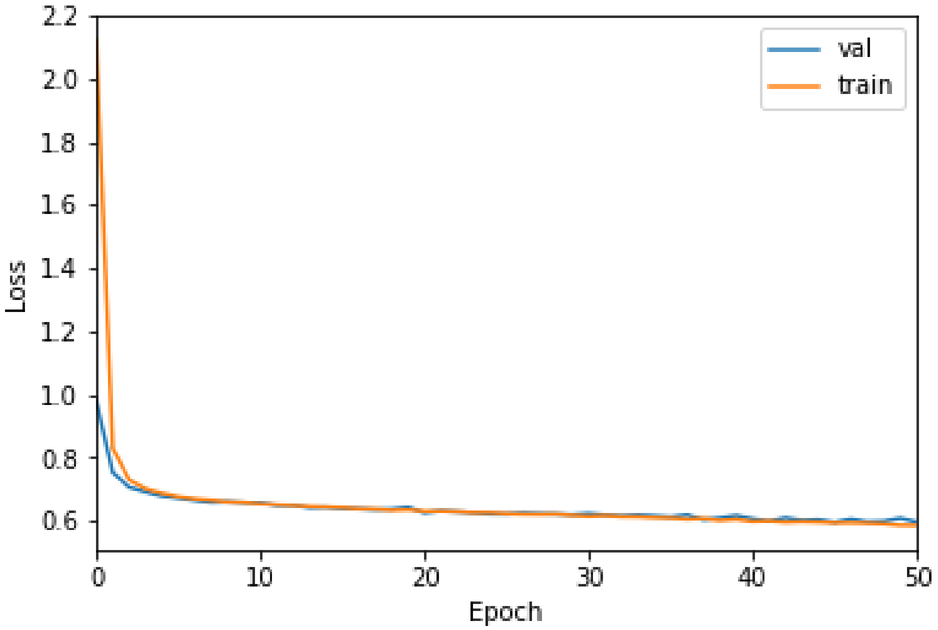


Figure 5.3.1: Plot of training loss curves for siamese network

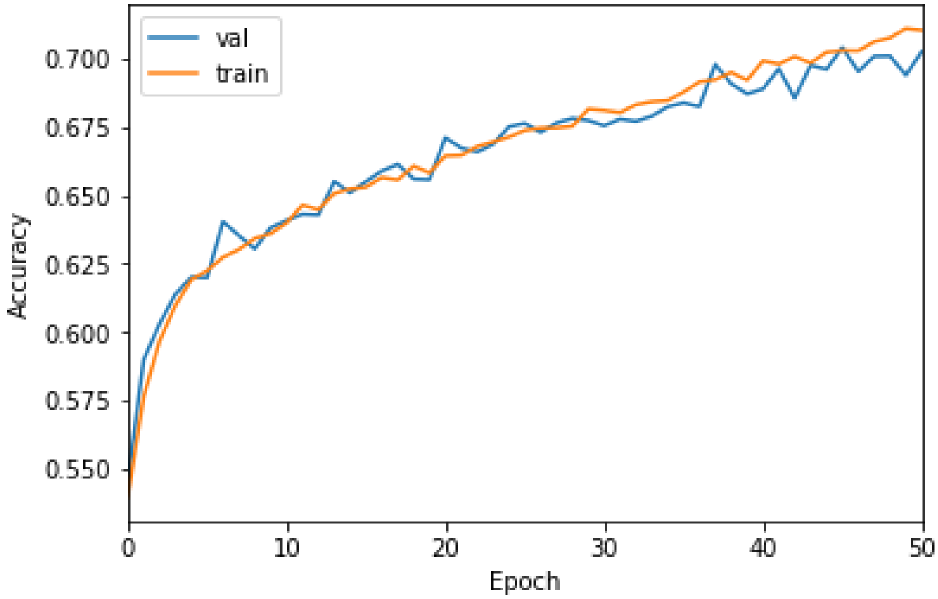


Figure 5.3.2: Plot of training accuracy curves for siamese network

On the validation set, the final model had an accuracy of 0.7005. For comparison, due to the balanced nature of our dataset, a completely random model (using the result of a fair coin flip to guess whether an influence relationship exists or not between two input songs) would have had an accuracy of approximately 0.5. Other metrics for the final model are summarized in the table below:

Table 5.3.1: Accuracy metrics for trained siamese network on validation set

Metric	Value
Accuracy	0.7005
Precision	0.6875
Recall	0.7353
F1 Score	0.7106

5.3.1 Same-Song Sample Prediction

As a sanity check, we also tested the model's accuracy by creating pairs of samples where both samples were from the same song. Despite not being trained on this task, the model had an accuracy of 0.9018 for identifying that the samples were from the same song.

5.3.2 Intergenre vs. Intragenre Prediction

We evaluated accuracy, precision and recall for intergenre v. intragenre prediction, using artist genre metadata information we gathered from AllMusic. Intergenre is defined as the two songs in the pair coming from artists of different genres (i.e. Jazz v. Classical) and intragenre is defined as sharing the same genre. The results are summarized in the table below:

Table 5.3.2: Accuracy metrics for intergenre vs. intragenre prediction

Metric	Same Genre	Different Genre
Accuracy	0.7134	0.6971
Precision	0.8484	0.4198
Recall	0.7676	0.6471
F1 Score	0.8060	0.5092

We see that across all metrics, the model outperforms on prediction when both artists are from the same genre vs. when the two artists are from different genres. In particular, we see that precision greatly suffers when the artists are from different genres.

5.3.3 Average Prediction Accuracy by Time Between Release Years of Songs

We also evaluated the average accuracy of our siamese network vs. the number of years between the release years of songs in input pairs:

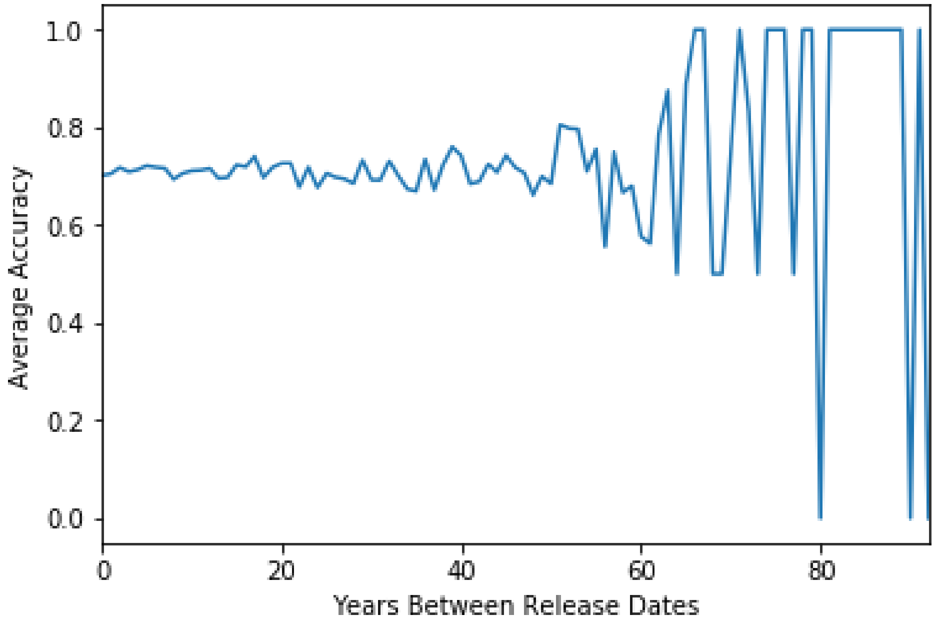


Figure 5.3.3: Plot of average accuracy vs. time between release years of songs

Note that the wide variance in the right-hand portion of the plot is due to the small number of samples that had a year difference greater than 60, especially when it came to positive examples of influence relationships. To test for a relationship between time between release years and model accuracy, we conducted a likelihood-ratio test (LRT) of a logistic regression model including time as a single predictor vs. a null intercept-only model, with a binary indicator for whether the siamese network was accurate as the response variable for both models. The LRT returned $p = 0.77$, so we failed to reject the null model and concluded that there was no statistically significant relationship between time between release years and model accuracy.

5.3.4 Qualitative Error Analysis

To get a better sense of how our siamese network was making errors (i.e. predicting influence relationships when there existed none), we filtered out for cases where the model predicted with very high probability (> 0.95) that there existed an influence relationship when there in fact was none. These cases, which we will refer to as *high probability errors* are outlined in the table below, with the respective members of the pair separated by a comma and each member written in the format "Artist Name - Song Name":

Table 5.3.3: Examples of high probability errors for siamese network

Input Pair	Predicted Probability
Brainiac - Hot Seat Can't Sit Down, Nirvana - Been a Son	0.985368
Buzzcocks - Fast Cars, Xasthur - Trauma Will Always Linger	0.976721
Babyland - Past Lives, Ramones - Gimme Gimme Shock Treatment	0.976054
Tony Iommi - Paranoid, Iron Maiden - Powerslave	0.973814
Nine Inch Nails - The Hand That Feeds, Badlands - Ride the Jack	0.973143
Winter - Winter, Neil Young - The Needle and the Damage Done	0.972291
Fatboy Slim - Right Here Right Now, Swans - I Am the Sun	0.970423
Jimi Hendrix - Machine Gun, Roger Miller - Old Friends	0.96569
Nirvana - Been a Son, Gorilla Biscuits - New Direction	0.96561
Ramones - Gimme Gimme Shock Treatment, Yves Deruyter - The Rebel	0.961905
Bo Diddley - Road Runner, Zaiko Langa Langa - Egide	0.960632
Leila Pinheiro - Renata Maria, Fatboy Slim - Right Here Right Now	0.960359
Corrosion of Conformity - Stare Too Long, KT Tunstall - Suddenly I See	0.95792
Skywave - Here She Comes, NOFX - Bob	0.951591
Hawkins Family - Changed, Lee "Scratch" Perry - Heavy Voodoo	0.95039

In the vast majority of these cases, both members of the input pair belong to the same genre of Pop/Rock, so this may partially explain why the model has difficulty, though in section 5.3.2 we did note that the model tends to perform better on intragenre prediction on aggregate, so this is not in line with that trend.

Listening to the audio samples themselves, the mistakes the model made seem reasonable for the most part. For instance, the two tracks in the first pairing — Brainiac - Hot Seat Can't Sit Down, Nirvana - Been a Son — do overlap acoustically. Both tracks feature male lead vocals with similar timbres, grungy guitar and a strong rock backbeat. In fact, given the proximity of the active years for the two bands (1992-1997 for Braniac and 1987-1994 for Nirvana), it is possible that one of the bands influenced the other even if that is not reflected in the AllMusic influence graph.

On the other hand, the pairing of Bo Diddley - Road Runner, Zaiko Langa Langa - Egide, which the siamese network predicts to be an example of an influence relationship with probability 0.95 appears rather out-of-place when listening to the two tracks alongside one another. The former is a 1960s 12-bar blues by an American musician while the latter is an upbeat-sounding 1995 dance number by a group from the Democratic Republic of the Congo. The closest sonic element that the two tracks appear to have in common is a similar tempo, but that is about it.

While we will not exhaustively go through each of these pairings, this sort of qualitative analysis does suggest elements that the network may be picking up on, such as timbre, groove, instrumentation and tempo, which are indeed some elements that a human listener would pay attention to as well. That said, this is to a certain extent speculative; the convolutional filters of the network could just as well be latching onto some other aspect of the mel-spectrogram representation not discussed here. Though there have certainly been recent developments in interpreting CNNs [15], at the moment there is simply no way to tell for certain what specific elements of the songs our model focuses on in generating predictions.

5.4 Model Application: Ranking Influence

One natural influence-related question one may ask is, given a collection of an artist's influencers (people who influenced the artist in question), how might we rank the relative importance of these influencers in terms of impact on that artist's music? Indeed, this question is particularly interesting in the case of the ground truth influence graph from AllMusic given that it only contains edges indicating influence relationships with no information about the relative *strength* of these relationships.

5.4.1 Ranking Algorithm Definition

We propose the following algorithm which applies our trained siamese network to answer the question of the relative ranking of influencers:

1. For a given artist u, get the set A_u of all influencers (ancestors) of u according to the ground truth graph. Therefore $\{(a, u) : a \in A_u\}$ would be the set of all directed edges terminating at node u in the ground truth graph.

2. Estimate the average probability of influence for an influencer-artist pair (a, u): create an input pair for the trained siamese network by randomly sampling a 3-second snippet of the respective mel-spectograms for each artist in the pairing and run the trained model to obtain a predicted probability of influence p_1^a where the superscript indicates the influencer we are considering and the subscript indicates which sample we are on. Independently sample a total of n times for this influencer-artist pair, so we have n influence probabilities $p_1^a, p_2^a \ldots p_n^a$. Take the mean to get an estimated average probability of influence for the influencer-artist pair $\hat{p^a} = \dfrac{\sum_{i=1}^n p_i^a}{n}$.

3. Repeat step (2) for all $a \in A_u$ to get an estimated average probability of influence for every influencer-artist pair.

4. Normalize the estimated average probabilities of influence for each influencer-artist pair to yield an *estimated influence proportion* for each influencer: $\hat{p}^a_{norm} = \dfrac{\hat{p}^a}{\sum_{a' \in A_u} \hat{p}^{a'}}$

Therefore in the end we obtain an estimated influence proportion for each influencer-artist pair $\hat{p}^a_{norm}$ with $\sum_{a \in A} \hat{p}^a_{norm} = 1$. A higher value of $\hat{p}^a_{norm}$ for a given influencer can be interpreted as the influencer being more influential on the artist u, and we consequently now have a method of ranking influencers.

5.4.2 Qualitative Analysis

To see this algorithm in action, we apply it to rank the influencers of J. Cole, a popular Rap artist and Charlie Parker, widely regarded as the greatest jazz alto saxophonist of the 20th century, using $n = 100$ (100 three second samples per each influencer-artist pairing). The rankings of the influencers in decreasing order of estimated influence proportion for both of these artists as determined by our algorithm can be seen in the tables below, accompanied by qualitative analyses from a musicological perspective:

Table 5.4.1: Influence proportions of J. Cole's influencers by our ranking algorithm

Influencer Name	Estimated Influence Proportion
2Pac	0.124236
Pharrell Williams	0.124146
Jay-Z	0.124084
Nas	0.122706
Clipse	0.116895
OutKast	0.108001
Eric B. & Rakim	0.101246
Pete Rock	0.0903422
Murs	0.0883447

Analysis: According to J. Cole himself, his favorite rappers as a child were

2Pac and Jay-Z [1], who appear as the number 1 and number 3 influencers respectively for him according to our ranking algorithm. In another interview, J. Cole stated that in order, his favorite rappers of all time were: 2Pac, Biggie, Nas, Jay-Z, and Andre 3000 (one-half of the group OutKast) [2]. Discounting the artists in this listing who do not appear in the graph we scraped from AllMusic, the relative ordering given personally by J. Cole of 2Pac, Nas, Jay-Z and OutKast corresponds very closely to the ordering given by our algorithm in the table above. Recalling the discussion of various definitions of influence posed in the introduction, it is perhaps important to note that in real life Jay-Z was J. Cole's first mentor and in fact Jay-Z's label was the first one J. Cole signed to. Given the close personal relationship between the two, it is therefore perhaps plausible that Jay-Z has had a slightly greater impact than Nas on J. Cole's music as well, which is what our algorithm would appear to suggest.

[1] http://www.musictimes.com/articles/11093/20140930/j-cole-talks-jay-z-tupacs-influence-career-watch.htm

[2] https://www.hotnewhiphop.com/j-cole-lists-top-5-rappers-recalls-worshipping-eminem-news.13210.html

Table 5.4.2: Influence proportions of Charlie Parker's influencers by our ranking algorithm

Influencer Name	Estimated Influence Proportion
Roy Eldridge	0.0669437
Louis Armstrong	0.0664712
Ben Webster	0.0659504
Coleman Hawkins	0.0641175
Benny Carter	0.0634215
Barney Kessel	0.0631203
Buster Smith	0.0627861
Lester Young	0.0623917
Art Tatum	0.0620452
Erskine Hawkins	0.0619891
Johnny Hodges	0.0617768
Don Byas	0.0614892
Illinois Jacquet	0.0609022
Count Basie	0.0594951
Jay McShann	0.059046
Jimmy Dorsey	0.058054

Analysis: At first glance, this ranking appears puzzling because the top 2 artists, Roy Eldrige and Louis Armstrong are trumpet players whereas Charlie Parker is a saxophone player. This seems strange since in jazz, usually (though not always) an artist's strongest influencers tend to be players of the same instrument. However, listening to the audio clip of Charlie Parker used for sampling reveals a plausible explanation: the audio clip is of the standard "A Night in Tunisia", which features the trumpeter Dizzy Gillespie playing the song's melody alongside Parker. According to AllMusic, Gillespie himself was influenced by Eldrige and Armstrong. Thus the first two entries of our ranking make more sense given that our trained siamese network has no mechanism by which we can dictate which instrument to focus on, and furthermore this suggests that our model is able to pick up on instrument-specific timbres such as trumpet. Looking further down the ranking from our algorithm, Lester Young and Buster Smith should perhaps rank higher given that they are mentioned by critics as clear influencers of Charlie

Parker [3], though they do both appear in the upper half of the ranking.

Our proposed algorithm for ranking influence is still preliminary and requires further validation. Though the suggested trends as discussed above for the two examples of J. Cole and Charlie Parker are interesting, they obviously are not definitive proof of the efficacy of the algorithm. One potential cause for concern for instance is that the estimated influence proportions tend to be fairly close to one another in magnitude. That said, as an application of our trained model, the algorithm does demonstrate the versatility of our approach to modeling musical influence through siamese networks.

5.5 Discussion

5.5.1 Comparison to Morton and Kim

Morton and Kim [14] used deep belief networks [10] as feature extractors from spectral representations of songs before using logistic regression for classification. They treated influence prediction as a multi-label classification problem with 10 total classes, using the top 10 most influential artists from AllMusic in terms of outdegree as the classes. They achieved an F1-Score of approximately 0.4, though it is important to note that their results cannot directly be compared with ours due to differences in problem setup.

In contrast, our system using siamese convolutional neural networks is arguably more general. Instead of having a fixed number of artists as possible labels, our model takes in as input a pair of samples of songs and returns a binary prediction for whether there exists an influence relationship or not. This allows for extension to prediction on pairs where neither artist was seen during model training and for applications such as influence ranking, as we saw in the previous section. Therefore our model is perhaps a step closer to being an influence discriminator in a more general sense.

[3] https://www.allmusic.com/artist/charlie-parker-mn0000211758/biography

5.5.2 Limitations

The primary limitation of our method perhaps is the size of the samples used in training (3 second clips as opposed to the full 30 seconds we had available). We simply did not have the computational resources to use longer samples and still have the model train within a reasonable amount of time. Our model seems to have performed well even despite the short length of the samples, and this is perhaps plausible when one considers that when a human adjusts a radio dial, he/she is often able to figure out within seconds what he/she is listening to and whether to switch to the next station. That said, since music operates on several structural timescales, there is without a doubt information loss from such a limited timescale that our model is unable to account for.

In terms of information loss, we also only sampled one of the 10 tracks that we scraped per artist, and then further sampled a 3 second segment from that in the creation of our training pairs. One question then (which we were unable to address) is, given multiple audio samples per artist, how do we choose which ones to use when generating training pairs? After all, even if there exists an influence relationship between two artists, this might not be necessarily reflected in every song pairing consisting of a sample from each of the respective artists. A related question is, even assuming that one has a sufficiently "good" feature representation of a song (e.g. extracted from a CNN), how does one go from song level summarizations to an artist-level summarization? Obviously, certain heuristics such as averaging come to mind, but is there a more robust way? All of these remain open questions.

Only hope that we kinda have left is music and vibrations
Lot a people don't understand how important it is, you know

"Mortal Man"- Kendrick Lamar

6 Conclusion

In this thesis we investigated modeling musical influence through data, settling on an audio content-based approach using 143,625 audio files and a ground truth human expert curated network graph of artist-to-artist influence consisting of 16,704 artists scraped from AllMusic.com. We first tackled this problem through a topic modeling approach, using the Document Influence Model to find a significant correlation with node outdegree in our ground truth graph. Due to a need for richer feature representation and a desire to classify artist-to-artist influence, we proposed a novel approach using siamese convolutional neural networks, achieving a validation accuracy of 0.7 on predicting binary influence between 3 second mel-spectogram samples from pairs of input songs. Our method perhaps represents the most general attempt at modeling musical influence to date; we make no assumptions about the definition of influence, having the model learn to discriminate influence based on labeled examples and our model is easily extensible to song pairs (and hence influence relationships) not seen in training as opposed to having a fixed number of class labels. Additionally our method is extensible for use in other musical influence related applications such as relative ranking of influence strength as shown by the ranking algorithm we proposed.

What else can our model be used for? From a knowledge discovery perspective, it could be used to discover new influence relationships between music artists in a data-driven way. From a practical perspective, given the massive volume of music available for listeners today through services such as Apple Music, Spotify and Pandora (to name a few), there exists a need for ways of cataloging and organizing it all. Influence perhaps represents one such way. For instance, one can reasonably imagine a music recommendation system that incorporates influence information in curating playlists for listeners. In addition, with appropriate feature representations, our approach could be generalized for modeling influence in other forms of media, such as speech audio or text.

Our research has also raised many more questions and reflects broader issues beyond the fairly narrow scope of our work. For example, what is the best way to represent audio data? Admittingly, as we saw in this thesis, bag-of-words representations can be quite limited. We found good performance with applying deep learning methods to extract features from intermediate mel-spectrogram representations, but recently there have been beginning attempts to utilize raw waveforms directly which may serve as an interesting research direction [9]. As another example, given examples of multiple songs for a given artist, how do we create an artist-level summary beyond just simple averaging? Reworded more generally (in natural language processing terms), given a corpus of multiple documents for an author, how might we create an author-level summary? These questions are not just limited to, and in fact extend well beyond the problem of modeling musical influence.

References

[1] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[3] Nicholas J Bryan and Gen Wang. Musical influence network analysis and rank of sample-based music. In *ISMIR*, pages 329–334, 2011.

[4] François Chollet et al. Keras. https://github.com/keras-team/keras, 2015.

[5] Nick Collins. Computational analysis of musical influence: A musicological case study using mir tools. In *ISMIR*, pages 177–182, 2010.

[6] Nick Collins. Influence in early electronic dance music: An audio content analysis investigation. In *ISMIR*, pages 1–6, 2012.

[7] Sean Gerrish and David M Blei. A language-based approach to measuring scholarly impact. In *ICML*, volume 10, pages 375–382. Citeseer, 2010.

[8] Manuel Gomez Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1019–1028. ACM, 2010.

[9] Yuan Gong and Christian Poellabauer. How do deep convolutional neural networks learn from raw audio waveforms?, 2018. URL https://openreview.net/forum?id=S1Ow_e-Rb.

[10] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[13] Brian McFee, Luke Barrington, and Gert Lanckriet. Learning content similarity for music recommendation. *IEEE transactions on audio, speech, and language processing*, 20(8):2207–2218, 2012.

[14] Brandon G Morton and Youngmoo E Kim. Acoustic features for recognizing musical artist influence. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*, pages 1117–1122. IEEE, 2015.

[15] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. doi: 10.23915/distill.00010. https://distill.pub/2018/building-blocks.

[16] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[17] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178. ACM, 2010.

[18] Uri Shalit, Daphna Weinshall, and Gal Chechik. Modeling musical influence with topic models. In *International Conference on Machine Learning*, pages 244–252, 2013.

[19] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[20] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in neural information processing systems*, pages 2643–2651, 2013.