Lyric-based Music Recommendation

Derek Gossi
Mathematics & Statistics
University of Nevada, Reno

Mehmet H Gunes
Computer Science and Engineering
University of Nevada, Reno

***Abstract*—**Traditional music recommendation systems rely on collaborative filtering to recommend songs or artists. This is computationally efficient and performs well method but is not effective when there is limited or no user input. For these cases, it may be useful to consider content-based recommendation. This paper considers a content-based recommendation system based on lyrical data. We compare a complex network of lyrical recommendations to an equivalent collaborative filtering network. We used user generated tag data from Last.fm to produce 23 subgraphs of each network based on tag categories representing musical genre, mood, and gender of vocalist. We analyzed these subgraphs to determine how recommendations within each network tend to stay within tag categories. Finally, we compared the lyrical recommendations to the collaborative filtering recommendations to determine how well lyrical recommendations perform. We see that the lyrical network is significantly more clustered within tag categories than the collaborative filtering network, particularly within small musical niches, and recommendations based on lyrics alone perform 12.6 times better than random recommendations.

I. Introduction

Due to the proliferation of music streaming and subscription services in recent years, there has been increasing interest in determining how various songs and artists are connected to one another, and ultimately to a given listener. The goal in analyzing a musical network in this context is often to recommend songs or artists to a person who has a list of songs or artists that they are already known to enjoy listening to. This is a difficult challenge in many ways, especially when one considers the unique subtleties present in musical expression. Often, preferences for a given listener span a wide range of genres and styles, making labeling the data solely in this manner insufficient in the recommendation task. Up to this point, the methodology that has been employed for the recommendation task in an industrial setting has been largely focused on linking listeners together by preferences using a collaborative filtering method, i.e. if user 1 likes songs A, B, and C, and user 2 likes songs A, B, and D, the recommender might recommend song D to user 1, and song C to user 2. However, this methodology is an indirect approach, as it does not use actual musical or lyrical content. In particular, collaborative filtering recommendation systems do not scale well to new or existing entries suffering from a lack of user ratings.

A growing area of the literature in musical analysis is focused on using audio and lyrical features to classify artists and songs [4, 5, 6, 7]. This methodology goes beyond user preference lists and attempts to find relationships amongst the songs and artists. This, ideally, would lead to stronger recommendation engines, as well as a more thorough understanding of the music. The idea is to find factors beyond genre that influence a given listener's probability of enjoying a song. These could be related

to tempo, mood, production style, use in a dance setting, or lyrical sentiment. While analysis of the full audio wave data has the most potential to improve recommendation systems, the success of existing algorithms is limited when given the task of classifying this complex unstructured data. Another option is to utilize song lyrics to connect various songs and artists, where lyrics are available. This presents its own unique challenges, as accurate lyrical analysis involves being able to decipher subtleties such as irony, hyperbole, and ambiguity. Even disregarding the complexities of lyrical analysis, the lyrics in conjunction with the music provide another layer of complexity. For example, a given set of lyrics over a slow and maudlin musical background may be open to a completely different interpretation than the same set of lyrics over a fast-paced and energetic musical background. However, even considering these challenges, lyrical analysis has proved in the literature to be a worthwhile endeavor.

Research in the area of lyrical analysis has grown in recent years with the increasing availability of large datasets to train algorithms. Work in this area has tended to focus on classification of lyrical content into categories such as mood using labeled training data [3]. More recent work has separated lyrical analysis from standard text analysis, by showing the importance of rhyme, repetition, and meter. However, the research beyond standard text analysis tools has been fairly minimal. The release of the Million Song Dataset (MDS), by far the largest dataset available to use in research of this type, has begun a new wave of analysis [2]. What has been missing from the literature is a complex network approach to lyrical analysis where the topology of the network is defined by artists linked together by lyrical similarity. This paper uses lyrical network and compares clustering methods on these networks. While certain genres such as pop and rock have been notoriously difficult to separate, even for human analyzers, with the aid of the MSD there may be new possibilities for genre clustering and defining the edges.

II. METHODOLOGY

Lyrics Dataset: Lyrical data is provided by the musiXmatch Dataset (MXD) [1], which provides lyrics for 237,662 tracks and 22,821 unique artists (implying 10.4 average number of tracks per artist in the dataset), which are each directly linked to the MSD. This is, of course, a small subset of the full MSD. The remaining tracks were omitted due to either copyright restrictions, a given track not containing lyrics, or duplication. The lyrics for each track are provided in a BOW format, and are stemmed using a modified version the Porter2 stemming algorithm. Thus, words that are similar in a statistical sense, e.g. kneel, kneeled, and kneeling, are mapped to the same stem (in this case, “kneel”), and are treated as identical terms in the analysis. The total dictionary of terms used as features in this analysis is limited to the top 5,000 words present in the dataset, which accounts for approximately 92% of the complete set of unique words. This limited dictionary is chosen because many of the term features outside this list are noisy and unusable, or used too infrequently in the dataset to be of much statistical value.

Term Frequency Matrix: A vector space model (VSM) is implemented to represent the lyrical data. While the BOW format loses valuable information about the location of words relative to each other within a given song, it is a convenient statistical tool as it allows us to define a song as a vector along a finite dictionary of

terms, where each component in the vector represents the frequency of a given word. Once each song is vectorized in this manner, we have a sparse term frequency matrix of size $n\ x\ 5000$, where $n = 237{,}662$ songs in the dataset. Further, we create a similar reduced term frequency matrix of artists by adding frequency vectors across a given artist's song catalog within the dataset, resulting in a sparse $n_a\ x\ 5000$ matrix, where $n_a = 22{,}821$ unique artists in the dataset. This "summed" artist matrix represents the full dictionary of words used by a given artist in the dataset.

Term Frequency-Inverse Document Frequency (TF-IDF) Weighting: The term frequency matrix of artists, while modeling the lyrical data of each artist as elements in a common vector space, suffers from the fact that "unimportant" and "important" words are weighted similarly. This is remedied by first eliminating statistically unimportant stop words such as *the, is, at, which,* and *on*. Secondly, term frequency-inverse document frequency weighting (TF-IDF) is utilized to minimize the importance of common words occurring frequently in the dataset. Formally, we multiply the raw term frequency by the inverse document frequency $IDF(w) = \log\left(\frac{|A|}{|a \in A : w \in a|+1}\right)$ where A is the set of all artists in the dataset, and $a \in A$. Calculating TF-IDF weights for each element in the term frequency matrix results in an adjusted sparse matrix in the same vector space as the term frequency matrix.

Pairwise Similarity Matrix: Pairwise similarity between artists is calculated using cosine similarity. If x and y are artist TF-IDF vectors, then their cosine similarity is $C\left(\underline{x}, \underline{y}\right) = \frac{\underline{x}\underline{y}^T}{\|\underline{x}\|\|\underline{y}\|}$. Computing the cosine similarity for each artist vector results in an $n_a\ x\ n_a$ pairwise similarity matrix where 0 implies the two vectors are orthogonal and completely dissimilar, and 1 implies two artists are identical, in terms of frequency and types of words used in the lyrics.

Threshold Selection: From the similarity matrix, we are able to define which pairs of artists are "similar" to each other in the lyrical sense. Rather than applying a similarity threshold directly to similarity matrix to obtain the set of edges, we utilize a *k nearest neighbor* approach to emulate what would be seen in a traditional collaborative filtering recommendation network. The k nearest neighbors of a given artist are the k artists which would be recommended to a user given a known preference to the artist in question. This fixes the outdegree for every node in the dataset at k, while the indegree is of unknown distribution. This resembles real-world limitations of recommendation networks, as explored in [3]. The level k is chosen to be 10 for this analysis, a level deemed sufficient enough to collect a useful range of connections while not going beyond the practical limitations of a real-world recommendation network. However, it should be noted that there is no true empirical justification for this choice of k.

Collaborative Filtering Network: We compare the topology of the network defined by lyrical similarity to that of a traditional collaborative filtering approach. The Echo Nest Taste Profile Subset (ENTPS) [1], provided as part of the Million Song Dataset Challenge, includes data on the number of times a given user has listened to a song in the MSD. The data includes 1,019,318 unique users with 48,373,586 user/song/count triples. With this dataset, a traditional collaborative

filtering network can be defined by utilizing memory-based filtering using songs rather than users, i.e. item-based collaborative filtering. We begin by vectorizing data for a given item—in this case a song—where the i^{th} component in the vector represents the number of plays by user i. We compute pairwise cosine similarity for each song in the dataset where the song vectors belong to the vector space of users. Once the pairwise similarity matrix is generated, we compute the edges of the network using the *k nearest neighbors* approach, with k=10, in a similar manner to the lyrics network. The lyrics network and the collaborative filtering network are then reduced to 18,290 unique artists shared by both datasets, implying 80.2% of the original 22,821 unique artists in the lyrics dataset also have user play count data in the ENTPS. With each node having outdegree of 10, each of the two artist networks has 18,290 nodes and 182,900 edges.

Tag Data: To enhance analysis of the recommendation networks, we link user-generated tag data to artists in the network by utilizing the Last.fm dataset [1]. The Last.fm data is linked to the MSD and includes 522,366 unique tags and 505,216 tracks with at least one tag. We significantly reduce the tag set to the most general and descriptive tag categories. The 500 top tags in the dataset are grouped into relevant tag categories representing a unique musical genre, mood, or gender. Table 1 presents the unique tags in the Last.fm dataset that are used as tag categories. Overall, 23 tag categories are considered. 15 of the 23 tag categories represent musical genre. Four mood categories are considered. For simplicity, several of the genre groups include groups of similar, but technically different, genres. For example, soul, R&B, and funk are grouped together, while in actuality they are distinct genres.

Subgraph Analysis: To determine how each network is structured within certain musical communities, subgraphs are generated and analyzed for each of the 23 tag categories. Subgraphs are generated by limiting the artist set within a given subgraph to only artists featuring a tag from the tag category associated with that subgraph. By analyzing the number of edges remaining in each subgraph compared to the number of edges "leaving" the subgraph, we can see if recommendations within each network tend to stay within certain tag categories, or certain niches within the full artist set. Formally, we compare the actual number of edges in each subgraph to the number of edges that would remain in the subgraph given that the 10 outgoing edges from each node in the subgraph are distributed to random nodes in the network. The ratio to the actual number of edges in the subgraph to the expected randomly distributed number of edges in the subgraph will indicate how much recommendations tend to remain in a given tag category. The expected number of edges remaining in the subgraph given random edge distribution is calculated as follows. Given every node in the graph has outdegree of k, say node s is inside the subgraph S of full node set N. Then the expected number of edges from s going to other nodes in S is $\sum_{i=1}^{k} i \frac{\binom{|S|-1}{i}\binom{|N\setminus S|}{k-i}}{\binom{|N|-1}{k}}$. If edges are chosen randomly, each $s \in S$ selects edges independently. Expected number of edges remaining in the subgraph S is $|S| \sum_{i=1}^{k} i \frac{\binom{|S|-1}{i}\binom{|N\setminus S|}{k-i}}{\binom{|N|-1}{k}}$. For large N, this is approximately $|S| k \frac{|S|}{|N|}$.

TABLE I. LAST.FM TAG CATEGORIES

Category	***Type***	***Unique Tags Included***
Rock	Genre	Rock, Classic R., Hard R., Progressive R., Pop R., Soft R., Rock n Roll
Pop	Genre	Pop, Pop Rock
Alternative	Genre	Alternative, Alternative Rock
Indie	Genre+	Indie, Indie Rock, Indie Pop
Electronic	Genre	Electronic, Electronica, Electro, House, Trance, Techno, Progressive Trance
Dance	Genre+	Dance, party, club
Jazz	Genre	Jazz, Jazzy
Folk	Genre	Singer-Songwriter, Folk, Acoustic, Folk Rock, Singer Songwriter,
Metal	Genre	Metal, Heavy M., Death M., Progressive M., Black M., Power M., Gothic M., Melodic M., Doom M., Thrash M., Metalcore, Nu M.
Soul	Genre	Soul, RnB, Funk, R&B, RB, R and B
Hip Hop	Genre	Hip-Hop, Hip Hop, Rap, Hiphop
Punk	Genre	Punk, Punk Rock
Blues	Genre	Blues, Blues Rock
Country	Genre	Country, Classic Country
Reggae	Genre	Reggae
Latin	Genre	Latin, Spanish, Latino
Christian	Genre	Christian, Worship
Relaxing	Mood	Chillout, Mellow, Chill, Relax, Relaxing, Calm, Chill Out
Romantic	Mood	Love Songs, Love Song, Sensual, Sex, Sexy
Positive	Mood	Fun, Happy, Upbeat, Energetic, Uplifting, Feel Good, Energy, Positive
Negative	Mood	Sad, Melancholy, Melancholic, Dark, Moody, Bittersweet
Male	Gender	Male Vocalists, Male Vocalist, Male Vocals, Male
Female	Gender	Female Vocalists, Female Vocalist, Female, Female Vocals, Female Vocal

Comparison of the Recommendation Task: We compare the ranked recommendation lists for both networks, assuming that the collaborative filtering network represents the "true" rankings. For each artist, we consider the ranked list of the top 1,000 most similar artists using both lyrical similarity and user similarity metrics, and calculate the difference between the two ranked lists used a Rank Biased Overlap (RBO) metric, calculated as follows. Let $CF_i^{(j)}$ represent the set of the first i elements in the ranked list of the collaborative filtering network for artist j. Let $L_i^{(j)}$ be defined in an identical manner for the lyrical network. The RBO of the lyrical and collaborative filtering rankings for artist j is $RBO^{(j)} = \frac{1}{1000}\sum_{i=1}^{1000}\frac{CF_i^{(j)} \cap L_i^{(j)}}{i}$. The mean RBO for the full artist set is the mean of the $RBO^{(j)}$ across all artists j. We also compare the RBO of the collaborative filtering rankings with a set of random rankings for each artist, to see if using lyrical similarity rankings is how much more accurate than simply recommending a set of randomly ordered artists in the dataset.

III. RESULTS

Comparing the general topology of the two networks in Table II, several differences stand out. The average clustering coefficient of the lyrics network is significantly greater than that of the collaborative filtering network and the average shortest path is also greater. This indicates that the lyrics network, when compared to the collaborative filtering network, tends to be clustered more around certain

communities of the network, which would also increase the average shortest path as there are less bridges between communities.

TABLE II. NETWORK TOPOLOGY COMPARISON

Network	*Diameter*	*Average Shortest Path*	*Clustering Coefficient*
Lyrics Network	10	4.52	0.217
CF Network	6	4.22	0.119

While the average outdegree of both networks is fixed at 10, the indegree distributions are displayed in Fig. 1 and Fig. 2. Neither network displays a power law in its indegree distribution, with clear curvature in the log-log plots. The lyrics network is significantly biased than the collaborative filtering network, with the top 10% of nodes receiving 65.1% of the possible edges. In comparison, the top 10% of nodes in the collaborative filtering network only receive 22.6% of the possible edges.

Fig. 1. Lyrics Network

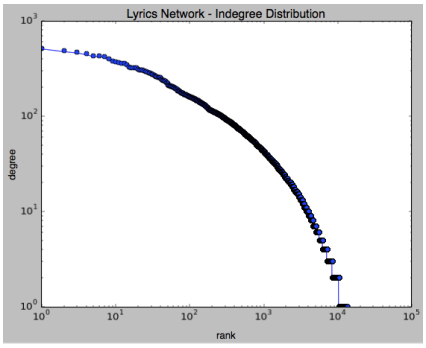
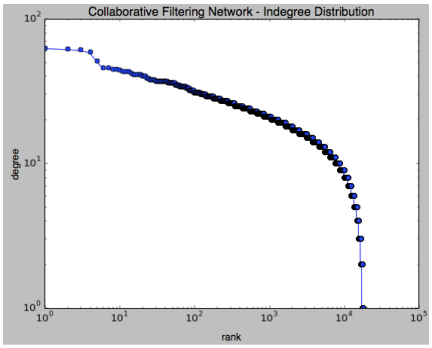


Fig. 2. Collaborative Filtering Network



Results of the subgraph analysis shown in Table III and Table IV indicate that the lyrics network is significantly more clustered within certain tag categories than the collaborative filtering network, indicating that users tend to listen to music across a broad spectrum of categories. However, users do not randomly listen to music across tag categories, as every tag category subgraph does have more edges than the expected number of edges given random edge preference. In addition, in both networks as the size of the node set in a subgraph decreases, the ratio of actual edges to expected random edges increases. This makes intuitive sense, as smaller tag categories indicates more specific niches—these specific niches would tend to have more unique lyrical sets, and listeners of a specific niche of music likely would not venture outside this niche as much as a listener of popular genres or categories.

In the lyrics network, Folk, Metal, Country, Hip Hop, Blues, Latin, and Christian music are particularly strong communities, as would be expected. In the collaborative filtering network, Metal, Country, Latin, Reggae, and Christian music display strong community preference, with Indie music also displaying above average community structure given the size of the artist set. In terms of mood category, negative music is more clustered than positive music lyrically, with ratios of 3.1 and 2.5, respectively, but neither is significantly clustered in terms of user preference. Music with male vocals is slightly more clustered lyrically than music with female vocals.

TABLE III. LYRICS NETWORK SUBGRAPH ANALYSIS

Subgraph	*Nodes*	*% of Nodes*	*Edges*	*Random Edges*	*Actual / Random*
Rock	7,622	42%	61,658	31,763	1.9
Pop	6,277	34%	45,870	21,542	2.1
Relaxing	5,425	30%	37,959	16,091	2.4
Alternative	5,126	28%	35,333	14,366	2.5
Positive	5,052	28%	34,657	13,954	2.5
Male	4,529	25%	27,533	11,215	2.5
Romantic	4,416	24%	30,058	10,662	2.8
Dance	4,404	24%	23,320	10,604	2.2
Indie	4,192	23%	23,987	9,608	2.5
Negative	4,135	23%	28,643	9,348	3.1
Female	3,617	20%	15,538	7,153	2.2
Folk	3,588	20%	22,603	7,039	3.2
Electronic	3,536	19%	12,981	6,836	1.9
Soul	3,123	17%	16,494	5,333	3.1
Metal	2,535	14%	14,377	3,514	4.1
Punk	2,282	12%	10,715	2,847	3.8
Hip Hop	2,088	11%	10,703	2,384	4.5
Jazz	1,955	11%	7,040	2,090	3.4
Blues	1,752	10%	8,128	1,678	4.8
Country	1,249	7%	6,094	853	7.1
Latin	1,117	6%	6,675	682	9.8
Reggae	818	4%	1,391	366	3.8
Christian	711	4%	1,960	276	7.1

TABLE IV. COLLABORATIVE FILTERING NETWORK SUBGRAPH ANALYSIS

Subgraph	*Nodes*	*% of Nodes*	*Edges*	*Random Edges*	*Actual / Random*
Rock	7,622	42%	39,382	31,763	1.2
Pop	6,277	34%	27,180	21,542	1.3
Relaxing	5,425	30%	20,880	16,091	1.3
Alternative	5,126	28%	20,561	14,366	1.4
Positive	5,052	28%	18,124	13,954	1.3
Male	4,529	25%	14,849	11,215	1.3
Romantic	4,416	24%	14,428	10,662	1.4
Dance	4,404	24%	14,198	10,604	1.3
Indie	4,192	23%	15,045	9,608	1.6
Negative	4,135	23%	12,928	9,348	1.4
Female	3,617	20%	9,764	7,153	1.4
Folk	3,588	20%	10,357	7,039	1.5
Electronic	3,536	19%	10,161	6,836	1.5
Soul	3,123	17%	8,159	5,333	1.5
Metal	2,535	14%	7,450	3,514	2.1
Punk	2,282	12%	5,711	2,847	2.0
Hip Hop	2,088	11%	4,598	2,384	1.9
Jazz	1,955	11%	3,280	2,090	1.6
Blues	1,752	10%	2,799	1,678	1.7
Country	1,249	7%	2,484	853	2.9
Latin	1,117	6%	1,921	682	2.8
Reggae	818	4%	901	366	2.5
Christian	711	4%	1,083	276	3.9

Table V displays a comparison of collaborative filtering recommendations with lyrical similarity recommendations utilizing the mean RBO metric. This is compared to the mean RBO of collaborative filtering recommendations and random recommendations to provide a baseline. While the lyrical recommendations have a weak mean RBO of 0.0649, it is 12.6 times superior to random recommendations.

TABLE V. RECOMMENDATION PERFORMANCE COMPARISON

Ranking Compared to CF	***RBO***	***Multiple***
Lyrical Ranking	0.0649	12.6
Random Ranking	0.0052	1

IV. CONCLUSIONS

When actual user data is unavailable, which especially holds true for many new and emerging songs or artists, it may be advantageous to consider content-based recommendation methods in determining the initial recommendations to and from this artist or song. This paper shows that even a purely lyric-based method provides significant information about the tags the artist or song might have associated with it. Lyrical analysis may be especially successful for niche genres such as Country, Metal, Blues, Hip Hop, and Christian—where the lyric network in this analysis was successful at differentiating these genres from others. At the very least, lyrical analysis could verify whether or not a given recommendation in the lack of user data is "very bad," as it could determine how far a user is venturing from their typical listening history.

By adding more elements to the vector space than just TF-IDF vectors for lyrics, such as direct factors for lyrical sentiment, repetition, and variety of word choice one could likely refine the results of this analysis. Adding factors for audio signal content would also produce stronger results.

Acknowledgements This material is based upon work in part supported by the National Science Foundation under grant number EPS- IIA-1301726.

V. REFERENCES

[1] T. Bertin-Mahieu, Daniel P.W. Ellis, B. Whitman, and P. Lamere. "The million song dataset," In Proceedings of the ISMIR, 2011.

[2] The Million Song Dataset Challenge. Kaggle Inc. www.Kaggle.com, 2012.

[3] P. Cano, O. Celma, and M. Koppenberger. "The topology of music recommendation networks," Universitat Pompeu Fabra, February 2008.

[4] X. Hu, J. S. Downie, and A. F. Ehmann. "Lyric text mining in music mood classification." University of Illinois at Urbana-Champaign, 2009.

[5] Y. Xia, K. Wong, L. Wang, and M. Xu. "Sentiment vector space model for lyric-based song sentiment classification." Association for Computation Linguistics, June 2008, pp. 133-136.

[6] T. Maxwell. "Exploring the music genome: lyric clustering with heterogeneous features," University of Edinburgh, 2007.

[7] R. Macrae, S. Dixon. "Ranking lyrics for online search," 13th International Society for Music Information Retrieval Conference (ISMIR 2012), pp. 361-366.

[8] F. Aiolli. "A preliminary study on a recommender system for the million song dataset challenge," University of Padova, Italy, 2012.