Content-Based Playlist Generation: Exploratory Experiments

Beth Logan
Hewlett-Packard Labs
One Cambridge Center
Cambridge Massachusetts 02142 United States
+1 617 551 7600
Beth.Logan@hp.com

ABSTRACT

We study the use of content-based approaches to form playlists from a given seed song. Our techniques use as a basis our previously presented audio similarity measure. This measure compares songs according to the novelty of their frequency spectrum and has been shown to have good performance on a non-trivial database. In this paper we investigate extensions to simply choosing the N closest songs to a seed as the playlist. Specifically, we study playlists formed by trajectories through the distance space and playlists formed using automatic relevance feedback. We report results on a database of over 8000 songs. We find that when information about the songs' genre is added, improvements over the basic distance measure are obtained, suggesting both approaches are suitable for incorporating user input or labeling information if available.

1. INTRODUCTION

The popularity of the MP3 compression format has changed the way people store, access and acquire music. It is now possible to carry hundreds of hours of music on a small device. Through the Web potentially millions of hours are ubiquitously available. This change in scale of accessible music from the traditional album to millions of songs raises many unanswered questions of how to efficiently access and discover this data and best present music to the user.

Ideally, we imagine a system which can automatically sense a user's mood or desires and play suitable music from a massive repository of available songs. The system would also respond to and learn from user feedback and be able to suggest suitable new songs from other repositories.

Although we are far from having such a system, researchers have made much progress toward this goal. In our lab, we are focussed on content-based analysis of music. We have previously developed a technique to quantify the similarity between songs based solely on their audio content [2], [3]. Our measure captures information about the novelty of the audio spectrum and therefore relates to the type of instruments playing. Thus we are concerned with 'genre' similarity rather than say melodic closeness. We previously found that this measure is useful for playlist generation, content-based copyright detection and music visualization.

In this paper, we focus solely on automatic playlist construction in which we desire to provide a user with a selection of music with a certain 'mood'. Previously, we chose a playlist as the N closest songs to a seed song according to our distance measure. In this paper, we consider more complicated schemes in which we post-process this initial list of songs. Specifically, we describe two extensions: tracing a trajectory through the distance space and using relevance feedback. Due to space limitations, we shall not describe our distance measure or many details of our experimental setup. The reader is referred to [2] for further information.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. © 2002 IRCAM - Centre Pompidou

2. GENERATING PLAYLISTS

In this section, we describe techniques which post-process initial playlists chosen as the N closest songs to the seed song.

2.1 Song Trajectories

We imagine a graph of all songs in our database. Each song is a node and links between songs describe how closely the songs are related. The simplest graph uses our distance measure for the link strength. A playlist can be formed using this graph by choosing the shortest path of length N emanating from the seed song.

If when tracing this path, a song is repeated implying a loop, we use one of two simple heuristics. In the first technique, if when choosing the closest song to s_n we encounter a loop, we chose the next closest song to s_n until we find a song that is unseen for this playlist. Thus we continue to expand the path from the *current* song. An alternative is to restart the path from the next closest unseen song to the *original seed*.

Graph-based playlists have been examined previously [1]. However, this prior work assumes that relevant attributes have already been determined for each song rather than extracting them from the audio as in our case.

2.2 Relevance Feedback

Relevance feedback is an established technique in the IR community (e.g. [4]). It aims to improve the quality of returned documents in response to a user's query by incorporating feedback from the user. A well-known related technique which does not require user input is automatic relevance feedback. Here, it is assumed that the top M documents returned are relevant. Features from these documents are then extracted and used to re-rank the documents in the collection, hopefully resulting in improved performance.

We have implemented a simple version of automatic relevance feedback for our music database as follows. Our scheme combines the simple playlists for the closest M songs to the seed song. Specifically, for a given seed song, we sum the distance scores of the simple playlists for the top M songs and use the resulting scores to form a final playlist. Table 2.2 illustrates this scheme.

3. EXPERIMENTS

We conduct experiments on an in-house database of over 8000 songs drawn from a wide range of styles. Each song in the database is labeled with the genre, song name, album name and artist name. The genres are assigned according to the *All Music Guide* (AMG) database (www.allmusic.com).

Table 1: Automatic relevance feedback used to form playlist from seed. Distances are shown in parenthesis.

N-Closest Playlists						Final	
Seed		Song1		Song2		Playlist	
Song1	(4)	Song2	(6)	Song4	(4)	Song2	(9)
Song2	(3)	Song4	(3)	Seed	(3)	Song4	(7)
Song3	(1)	Song3	(2)	Song1	(1)	Song1	(5)
...		...		...		...	

Our experiments examine playlists for all songs in the database and report the average number of relevant songs for playlists of length 5, 10 and 20. We use objective definitions of relevance - songs of the same style, by the same artist and on the same album - in order to conduct automatic tests over the whole database since user tests are beyond the scope of this paper. We previously found good correlation between subjective and automatic tests [2].

3.1 Baseline System

Table 2 shows the average quality of playlists in the baseline case when the playlist is simply the N closest songs to the seed song. We see that that on average, the majority of songs chosen are of the same genre as the query song and that playlists of size 5 contain one song by the same artist or on the same album. Note that these results give only an indication of performance. For example, several of our genre categories overlap (e.g. *jazz* and *blues*) and songs from both categories might still be perceived as relevant by a human user.

Table 2: Results for baseline playlists.

Relevance	Average nr. of relevant songs in playlist		
	Size 5	Size 10	Size 20
Same Genre	3.46	6.60	12.6
Same Artist	1.34	2.07	3.01
Same Album	1.11	1.63	2.21

3.2 Song Trajectory Playlists

The top part of Table 3 shows results for playlists formed from song trajectories. We show results for both variations discussed in Section 2.1. The results show that the technique of tracing paths though the song space gives worse results than the baseline. The second variation is somewhat better than the first however.

Table 3: Results for proposed extensions

Relevance	Scheme	Average nr. of relevant songs in playlist		
		Size 5	Size 10	Size 20
Same Genre	Trajectory,1	3.26	6.13	10.75
Same Artist		1.08	1.43	1.68
Same Album		0.89	1.11	1.22
Same Genre	Trajectory,2	3.33	6.37	12.08
Same Artist		1.23	1.89	2.73
Same Album		1.01	1.49	2.00
Same Genre	Feedback	3.40	6.54	12.46
Same Artist		1.27	1.96	2.83
Same Album		1.05	1.54	2.07

3.3 Automatic Relevance Feedback

The second part of Table 3 shows results for automatic relevance feedback as described in Section 2.2. We show results for the best M, $M == 1$. Comparing these results with the baseline results in Table 2, we see that using automatic relevance feedback results in slightly worse performance than the baseline.

3.4 Incorporating More Information

Our results for the trajectory and relevance feedback extensions to our basic distance measure are disappointing. Investigation of the playlists formed revealed that a major problem was 'tangential' songs. Because our distance measure is not perfect, expanding a path or combining scores of playlists from bad songs can corrupt the new playlist with irrelevant songs.

We therefore investigate whether adding information from labels or user input is beneficial. For the trajectory scheme, we experiment with the second variant. When we detect a loop and restart the playlist from the next closest unseen song to the seed song, we constrain this song to be from the same genre as the seed. We use genre to simulate a user highlighting songs of the same style. Results using this scheme are shown in the first half of Table 4. Similarly, we investigate a version of automatic relevance feedback where when choosing M songs to expand, we choose songs from the same genre as the seed song. These results are shown in the lower half of Table 4. Again we only show results for the best scheme which was $M == 5$.

Table 4: Results when genre information added.

Relevance	Scheme	Average nr. of relevant songs in playlist		
		Size 5	Size 10	Size 20
Same Artist	Trajectory	1.40	2.33	3.58
Same Album		1.16	1.82	2.60
Same Artist	Feedback	1.60	2.31	3.09
Same Album		1.30	1.79	2.24

The results in this table are encouraging since there is improvement in the Same Artist and Same Album metrics over the baseline in Table 2. This suggests that our new approaches provide a framework in which labeling information can be incorporated into the original distance measure. Also, we believe this could allow user input to be incorporated into playlist construction.

4. CONCLUSIONS AND FUTURE WORK

We have investigated the use of content-based techniques to form playlists from a given seed song. We explored two extensions to our previously published technique which simply chose the N closest songs to a seed [2]. The first extension forms playlists as trajectories through the distance space. The second uses automatic relevance feedback.

We evaluated our techniques on a database of over 8000 songs of varied styles. Surprisingly, the proposed extensions did not perform as well as simply choosing the N closest songs to the seed song as the playlist. We attribute this to the imperfect nature of our distance measure. However, when information about the songs' genre is added, improvements are noted, suggesting both approaches provide a framework for incorporating user input or labeling information.

5. REFERENCES

[1] M. Alghoniemy and A. H. Tewfik. A network flow model for playlist generation. In *ICME 2001*, Tokyo, Japan, 2001.

[2] B. Logan and A. Salomon. A content-based music similarity function. Technical report, Compaq Cambridge Research Laboratory, June 2001.

[3] B. Logan and A. Salomon. A music similarity function based on signal analysis. In *ICME 2001*, Tokyo, Japan, 2001.

[4] K. Sparck-Jones and P. Willet, editors. *Readings in Information Retrieval*. Morgan Kaufmann, 1997.