Large-scale Crossmedia Retrieval for Playlist Generation and Song Discovery

Filipe Coelho
Laboratório SAPO/U.Porto
Universidade do Porto
filipe.coelho@fe.up.pt

José Devezas
Laboratório SAPO/U.Porto
Universidade do Porto
jld@fe.up.pt

Cristina Ribeiro
DEI–FEUP & INESC TEC
Universidade do Porto
mcr@fe.up.pt

ABSTRACT

To explore vast collections of audio content, users require automated tools capable of providing music search and recommendation even when faced with large-scale collections. Collaborative-filtering recommenders rely on user-generated information and may be hindered by the lack of users or a bias for certain popular genres, enclosing users in an information bubble. Audio content analysis, on the other hand, is a reliable source of audio similarity, used in tasks such as music classification. For highly interactive tasks, however, the performance of analysis algorithms becomes an issue.

In this work, we address the playlist generation and song discovery tasks on large-scale datasets. We generate playlists and explore the collections with example-based queries using audio features, lyrics and tags. Approximate indexing and cross-media reranking are used for efficiency. Audio content is mapped to textual representations that can be handled by information retrieval libraries.

We explored the feasibility of this content-based approach in the Million Song Dataset, a large-scale collection of audio features and associated text data comprising almost 300 GB of information. The proposed strategy can be used independently as a content-based music retrieval system and as a component for hybrid recommender systems.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*abstracting methods, indexing methods*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*clustering, information filtering, search process, selection process*

General Terms

Performance, Human Factors, Experimentation

Keywords

Music search, song discovery, playlist generation

OAIR'13 May 22-24, 2013, Lisbon, Portugal.

1. INTRODUCTION

Social media networks introduced a shift to explore user provided information about tastes instead of analyzing content directly, in an effort to ease hardware requirements by taking advantage of data sparseness. However, the consequence of being based on user feedback, either in the form of implicit (usage) or explicit (ratings) data, is that recommender systems solely depending on collaborative-filtering algorithms suffer from the "cold start" problem, that is, the initial lack of user information. The system cannot model its users tastes without enough ratings and therefore recommend similar items. Also, by ignoring additional information, this type of recommender systems has the tendency to lock users into their tastes without suggesting other potential items that users might like but don't know yet. While computational problems can be solved with sparse user data, diversity and exploration of potential items are areas that can greatly improve user experience and satisfaction with recommender systems and that can highly benefit from content-based information to unlock users from their taste boundaries.

In this work, we address the task of music retrieval [1], specifically the subtasks of playlist generation [3] and song discovery. We follow a content-based approach adapted for large-scale datasets that has been proved useful in the related task of automatic text illustration [2]. An issue with content-based retrieval systems has been the complex extraction and indexing of multimedia features, requiring a huge amount of processing power in order to analyze and search this data. Our strategy addresses the issue with approximate indexing and crossmedia retrieval techniques. This approach can serve both as a standalone music retrieval system and as the basis for a hybrid recommender system combining content-based and collaborative filtering information.

Our vision is represented in Figure 1. This approach starts from a text-based query where users provide keywords, song titles, artist names or even lyrics excerpts. The resulting playlist groups clusters of similar songs, essentially pulling the most similar among themselves to the top and placing the "outliers" at the end of the playlist. Users can also pick a song from that playlist and reorder by closeness. Each song is then followed by the most similar to it that hasn't been already included. Lastly, to address song discovery, users can choose a specific song and search for similar ones over the entire collection by using its audio properties, or lyrics combined with tag information.

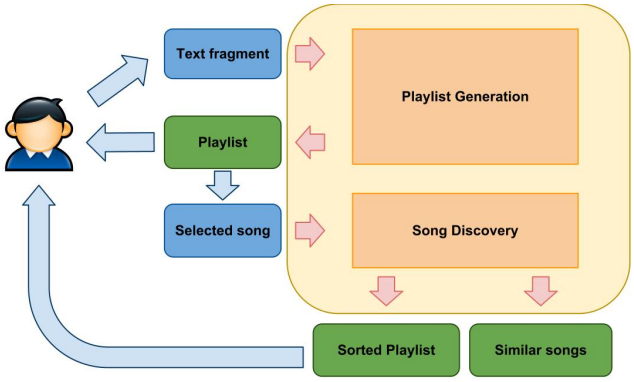

Figure 1: Music recommendation.

2. THE MILLION SONG DATASET

The Million Song Dataset[1] (MSD), released in 2011, is a freely-available collection of audio features and metadata for a million contemporary popular music tracks [4]. This multimedia dataset represents a significant step in this area, with the objective of encouraging research on large-scale algorithms, provide a reference evaluation dataset and help new researchers in Music Information Retrieval. While not having the original raw audio files, it provides the feature analysis and metadata from The Echo Nest[2].

2.1 Additional metadata and audio features

In this work we have also used two additional datasets provided with the MSD in order to obtain textual data for songs, similarly to recent works [5, 6, 7].

The Last.fm dataset, the official song tags and song similarity collection for the Million Song Dataset, covers 94% of the MSD tracks, with over half having at least one tag. While we disregarded the song similarity information, as it was based on user feedback, we took advantage of the available tags, using them as a textual feature to characterize our songs.

The musiXmatch dataset, the official lyrics collection for the Million Song Dataset, matches 77% of the MSD collection and provides lyrics in a bag-of-words format, due to licensing restrictions. Given the existence of genres with little or no available lyrics, this dataset actually covers "only" 24% of the MSD collection, which nevertheless is a substantial amount of useful information for text retrieval.

As for raw audio information, we have chosen specific features from those available in the MSD to represent each song: *time signature*, *tempo*, *mode*, *loudness*, *key*, *duration*, *pitch coefficient average* $\times$ 12, and *timbre coefficient average* $\times$ 12.

Table 1 shows some statistics about the dataset used. We could not successfully generate audio descriptors from all available songs due to missing data and we also chose to disregard mismatch data.

3. CROSSMEDIA RETRIEVAL

Recent work on handling large-scale datasets has used spatial trees with success [11], but here, instead of building specialized indexes for descriptors, we state it is possible to take advantage of the efficiency of textual indexing by mapping audio features to a textual form and indexing them with textual search engines, as previously done for image retrieval [10, 12]. Song metadata is pre-processed and stored in an inverted index. Audio low-level features are analyzed and used to build audio feature vectors, which are then transformed into a textual representation designated as *Surrogate Text Representation* (STR) [10]. These representations are handled in a common index and provide the means to search songs by similarity on both audio and textual features, including existing metadata and lyrics.

[1] http://labrosa.ee.columbia.edu/millionsong/
[2] http://the.echonest.com/

Table 1: The Million Song Dataset.

Songs	Size	Artists	Tags	Lyrics Terms
961,493	262 GB	44,263	214,809	4,920

3.1 Building surrogate representations

Content-based image similarity with global descriptors can be performed through an exhaustive linear search over the entire collection by comparing the query vector with every other. This strategy, while straightforward, is highly inefficient from a scalability perspective. Therefore, the need for effective approximate search has resulted in methods aimed at reducing the search space to a small number of potential matching candidates.

One of these methods involves the use of a small number of randomly chosen reference points, designated as "pivots" [8, 2]. Every song is compared to these pivots during the offline indexing phase, resulting in a ranked similarity vector that replaces the original feature vector during search.

In order to translate this approximate similarity to an STR usable by text search engines, an identifier (id) is assigned to each pivot. A pivot similarity vector for a song has the ids of a set of pivots in the order of decreasing similarity between the pivot and the song. For a given ranked pivot similarity vector with size P, its corresponding STR is built by appending each id $P - R + 1$ times, where R represents the rank of that pivot. As an example, for a song with a vector [B, C, A], which is closest to pivot B and farthest from A, the resulting STR is "B B B C C A". When analyzed by a search engine, this STR will carry the weight of each pivot, used when calculating the similarity between songs [9].

We do a mean-threshold transformation on feature vectors, transforming them into binary vectors to further improve performance. We determine the mean value of each feature over the entire dataset, obtaining a mean vector. We then compare each feature vector with the mean vector, assigning 0 to lower or equal values, and 1 to higher values. In this particular case, the resulting binary "hash" can be stored as a 32-bit integer. We use the Hamming distance to compare these hashes by counting the different bits. This reduces a costly distance operation to a simple exclusive-or that can be performed much faster than the Euclidean distance calculation.

We have empirically established the following parameters: for a collection of D unique binary vectors, $P = \sqrt{D}$ pivots are randomly chosen. This step is performed 10 times, selecting the group of pivots with greater internal distance (sum of the distances of each pivot to every other) as to maximize spectrum coverage. Every song is compared to

Table 2: Initial playlist.

Query: "coldplay live"	Score
See You Soon (Live In Sydney) - Coldplay	5.1
Shiver (Live In Sydney) - Coldplay	5.1
One I Love (Live In Sydney) - Coldplay	5.1
Amsterdam (Live In Sydney) - Coldplay	4.2
You Only Live Twice (Live Norway) - Coldplay	4.0
Daylight - Coldplay Tribute	3.9
Moses (Live In Sydney) - Coldplay	3.7
Yellow (Live In Sydney) - Coldplay	3.4
Speed Of Sound (Live) - Coldplay	3.2
Fix You (Live) - Coldplay	3.2

these pivots and the resulting ranked vector is trimmed to contain only the first $\sqrt{P}$ pivots. Finally, the global STR is generated and stored in a database for later indexing.

4. APPLICATIONS

Given the subjective nature of playlist generation and music discovery retrieval tasks, we present results from some predefined queries in order to demonstrate the usefulness of the content-based approach. There is extensive evaluation research on music retrieval [13, 14, 15], with user studies considered for a later stage in order to validate the approach of the experiments.

4.1 Playlist generation

Given a text query by the user, be it the name of a song or artist, a lyrics excerpt or emotion tags such as "happiness" or "betrayal", the system performs a textual search over the indexed fields of each song. This results in an initial playlist with no more than twenty songs. A subset of ten, for the "coldplay live" query, is shown in Table 2. "Score" refers to the Lucene text retrieval score. In this example, users wanted live performances from the Coldplay band, but they could also insert parts of songs or tags describing their mood or a specific music genre.

We apply a content-based rerank on the feature vectors of this initial playlist, based on the audio similarity between songs. Using the Hamming distance, we sum the distances between songs. Songs with a smaller total distance value, that is, with greater similarity to all the others, become more "central" in the playlist.

This method finds a parallel in the graph theory measurement known as "closeness centrality" [16]. Figure 2 illustrates the idea through a similarity graph, where weighted edges are defined for each pair of songs based on their distance. We state that tracks that are more similar between them represent a "cluster" of songs that may appeal the user, while "outliers" will be pulled to the end of the playlist.

Another option is to pick a song and reorder the list "by closest". We add the initial chosen song to an empty list, and the following song becomes the closest song that's not already on the list. By using this option, we allow the user to start with a favorite song and progress through the playlist with minimum disruption, as each next song is the closest to the current one.

4.2 Song discovery

The main advantage of the content-based approach is its user-independent nature, not affected by item popularity.

As an example, by using the live performance of the "One I Love" song, we were able to also retrieve the original version using the audio-only query, which indicates that even with the approximate indexing algorithm and binary hashing scheme, the audio features used in this work do capture meaningful information.

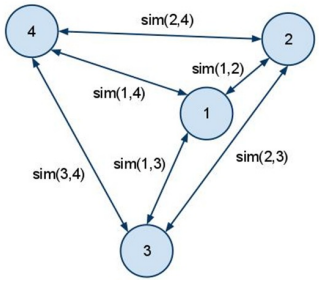

Figure 2: Graph-based rerank.

From a system performance point of view, the playlist generation and song discovery tasks are performed in seconds using the Lucene library. The full index, with metadata and audio features, takes over 12 GB and is not fully loaded to memory. Instead, Lucene handles its cache and retrieves only documents that are considered relevant for the query.

For even larger-scale collections, it is possible to split the index and content information between Lucene and a separate database, loading the index to RAM and retrieving data for visualization in a separate process.

5. EVALUATION

We take advantage of The Echo Nest Taste Profile Subset to evaluate the playlists generated by our system for a selected set of queries. We assume that a playlist is as good as the number of songs that co-occur in any of the user's libraries. Thus, we evaluate the quality of a playlist by calculating a score that accumulates the number of song pairs in the playlist that are played by at least a common user. Thus, given a playlist of n songs, the maximum score will be $n(n-1)/2$.

We generate twenty playlists of $n = 20$ songs each, based on the queries displayed in Table 3. For each playlist, we calculate the evaluation score described above. The higher this score, the more likely our system is to generate playlists whose songs a person would listen to together.

The obtained results demonstrate that text queries generate playlists containing more pairs of listened-together songs. Queries using both lyrics and tags obtain intermediate scores, and audio feature-based queries achieve low scores, but still above zero. This shows that context information is the best solution to retrieve an initial acceptable playlist, but audio features can still find similar songs according to our evaluation scheme. This information can be used to deliver balanced recommendations, considering the "safety" of context-based search with the potentional serendipity of content-based search, that is, the ability to discover music that would not be found by merely looking at metadata.

6. CONCLUSIONS

We have applied a recent strategy for automatic text illustration to the music retrieval area, namely the automatic generation of playlists and the discovery of similar songs

Table 3: Evaluation of the playlists.

Playlist	Query	Score_{text}	Score_{audio}	$\text{Score}_{tags/lyrics}$
1	coldplay live	0.7316	0.0000	0.1368
2	metallica slayer heavy metal	0.3526	0.0737	0.4105
3	nirvana days of the new grunge alice in chains	0.3474	0.0053	0.2842
4	jason mraz i'm yours	0.7263	0.0263	0.4684
5	happy good vibe	0.0105	0.0368	0.0842
6	sad depressing doom dark	0.0368	0.0421	0.3947
7	britney spears rihanna madonna	0.4105	0.0579	0.1158
8	norah jones diana krall jamie cullum	0.7632	0.0053	0.3158
9	miles davis john coltrane classic jazz	0.0316	0.0263	0.0526
10	frank sinatra new york	0.0368	0.0000	0.0947
11	bob marley reggae summer happy positive	0.2421	0.0789	0.2263
12	pop rock avril lavigne	0.4000	0.0474	0.0263
13	indiana jones soundtrack	0.0842	0.0105	0.3211
14	led zeppelin the who classic rock	0.1000	0.0053	0.2053
15	rockabilly 50s elvis presley	0.0000	0.0158	0.0421
16	country bluegrass bill monroe banjo	0.0158	0.0105	0.0368
17	dubstep skrillex new beat	0.1737	0.0263	0.0947
18	electronic aphex twin creative	0.0632	0.0000	0.0947
19	house techno trance bestof	0.0474	0.0105	0.1421
20	blues muddy waters robert johnon jimi hendrix	0.0684	0.0263	0.0316
Mean		0.2321 ± 0.2577	0.0253 ± 0.0241	0.1874 ± 0.1401

based on audio and tag information. We demonstrate that it is possible to obtain results in a few seconds even when searching over a large-scale dataset with one million songs.

Also, since the system uses only content-based information, there is no risk of becoming overfitted to the users preferences, implicitly inducing diversity. Future evaluation will involve user studies and song similarity information from Last.fm in order to further validate our perspective.

7. REFERENCES

[1] C. Liem, M. Müller, D. Eck, G. Tzanetakis, and A. Hanjalic: "The Need for Music Information Retrieval with User-centered and Multimodal Strategies," *Proceedings of the International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies*, 2011.

[2] F. Coelho, and C. Ribeiro: "Automatic Illustration with Cross-media Retrieval in Large-scale Collections," *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, 2011.

[3] B. Fields: "Contextualize Your Listening: The Playlist as Recommendation Engine," *PhD Thesis, Department of Computing Goldsmiths, University of London*, 2011.

[4] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere: "The Million Song Dataset," *Proceedings of the International Society for Music Information Retrieval Conference*, 2011.

[5] M. McVicar, T. Freeman, and T. De Bie: "Mining the Correlation Between Lyrical and Audio Features and the Emergence of Mood," *Proceedings of the International Conference on Music Information Retrieval*, 2011.

[6] R. Ferrer, and T. Eerola: "Looking Beyond Genres: Identifying Meaningful Semantic Layers from Tags in Online Music Collections," *Proceedings of the International Conference on Machine Learning and Applications Workshops*, 2011.

[7] D. Liang, H. Gu, and B. O'Connor: "Music Genre Classification with the Million Song Dataset," *Technical Report*, Carnegie Mellon University, 2011.

[8] G. Amato and P. Savino: "Approximate Similarity Search in Metric Spaces Using Inverted Files," *Proceedings of the International Conference on Scalable Information Systems*, 2008.

[9] C. Gennaro, G. Amato, P. Bolettieri, and P. Savino: "An Approach to Content-Based Image Retrieval Based on the Lucene Search Engine Library," *Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science*, 2010.

[10] G. Amato, P. Bolettieri, F. Falchi, C. Gennaro, and F. Rabitti: "Combining Local and Global Visual Feature Similarity Using a Text Search Engine," *Proceedings of the International Workshop on Content-based Multimedia Indexing*, 2011.

[11] B. McFee, and G. Lanckriet: "Large-Scale Music Similarity Search with Spatial Trees," *Proceedings of the International Society for Music Information Retrieval Conference*, 2011.

[12] F. Coelho, and C. Ribeiro: "Image Abstraction in Crossmedia Retrieval for Text Illustration," *Proceedings of the European Conference on Information Retrieval*, 2012.

[13] M. Slaney, and W. White: "Measuring Playlist Diversity for Recommendation Systems," *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, 2006.

[14] L. Barrington, R. Oda, and G. Lanckriet: "Smarter than Genius? Human Evaluation of Music Recommender Systems," *Proceedings of the International Symposium on Music Information Retrieval*, 2009.

[15] B. Fields, C. Rhodes, and M. d'Inverno: "Using Song Social Tags and Topic Models to Describe and Compare Playlists," *Proceedings of the Workshop On Music Recommendation And Discovery*, 2010.

[16] L.C. Freeman: "Centrality in social networks: conceptual clarification," *Social networks*, Vol. 1, No. 3, pp. 215–239, 1979.