Abstract

Although, nowadays there are several applications that allow managing music collections, they only support search by artist, album or genre. On the other hand there are applications that allow the exploration of historical playback, but do so mainly from a statistical perspective, and not with the aim of exploring music collection.

The visualization and exploration of music collection and the exploration of listening histories are two areas that claim, respectively, to propose solutions for the exploration of music collections and for the exploration of history. Although there are various solutions for each of the areas so far none of them tried to connect the listening histories to music collection.

Here we propose a solution for exploring music collections based on listening histories, allowing users to rediscover their music collection for every hour of the day. Through the ordering of the songs from the music collection using the information extracted from the history playback for each hour, we get the information needed to make a division into two categories (recommended songs and music from the history). The two categories are used to divide the interface into two areas: recommended songs and history songs.

The evaluation of our solution show that users acquire a new perspective of their collection and come back to listen to songs that were forgotten or were not heard in a while.

1. Introduction

Nowadays, the proliferation of technological devices and the amount of information we have access to, make demand for management data applications stiffer. Particularly in the field of music, we tend to store in our digital devices an increasingly significant amount of music. It is due to this high amount of information that our problems begin. How often have we wished to have a personal assistant who knew our musical preferences, and based on this knowledgeable, to present our collection of songs in a layout that takes into account the music we love at a certain time of the day?

Nowadays it is easy to find an application to store and organize the songs we have in digital format. However, these applications are typically not characterized and are impersonal. These easily catalog the tracks from our library by genre, artist, album, but do not go beyond these traditional types of organization. On the other hand, several applications where it is possible to make a graphical analysis of the user’s likes, based on the historical reproduction of musical tracks, begin to emerge. Thus a user can learn more about their musical tastes, and even identify hearing patterns.

In this work we intend to combine the visualization and exploration of a music collection with listening histories. Our main objective is the way we view and explore our collection. We want to allow users to rediscover their music collections for every hour of the day, through their listening history. To this end we present recommendations for songs that have not been heard for some time but that are in accordance with the musical tastes of the user for that time of the day.

In this paper we begin by identifying the research areas studied to make our solution and indicate those that contributed to our work. Then we explain the designed solution and present the evaluation carried out and the results obtained. Finally, we present the conclusions we reached and a proposal for future work.

2. Related Work

To achieve the goals we set ourselves, we studied two areas of knowledge: the visualization and exploration of music collections and the visualization and exploration of listening histories.

With the area of visualization and exploration of music libraries we intend to study the various forms
of representation and exploration of a music collection, including the techniques used to get the view and its limitations. With area of the listening histories, we intend to study the important data to retrieve from the historical data, its influence and the existing visualization techniques. By combining the two areas of expertise, we intend to make our solution based on the analyzed work.

2.1 Visualization and exploration of personal music collections

The area of visualization and exploration of music collections, is composed by several visualization techniques. Within the studied techniques we highlight these techniques: TreeMaps, graphs and clusters.

In the visualization technique TreeMaps we highlight the work done by Dias et al. [1], which describes an approach for exploration and visualization of music libraries with a large number of tracks. The presentation of search results is performed using the ordered semantic TreeMaps, where the size of the rectangles are proportional to the size of the music tracks that it contains (see Fig. 1).

![Fig. 1 Muvis visualization](image1)

Through the audio analysis from the information collected it is extracted the information necessary to calculate the similarity between rectangles in order to arrange them on the display. The data can be filtered according to the criteria chosen by the user: genre, artist, time, year, etc.

In the graph display techniques we highlight the work of Gouyon et al. [2] where they propose a solution to visualize and explore music and the similarity between them, based on a graph (see Fig. 2). To construct the graph it is taken into account the similarity between artists and their popularity in the Last.fm web service where besides popularity, are drawn all the necessary information regarding the artists. The similarity between songs is identified by the size of the graph edges. The larger they are, the less similar are the artists. The popularity of artists is identified by the size of the rectangles in which they are inserted. The interface is complemented with information on each artist, which it can be accessed by clicking on the intended artist.

![Fig. 2 Global visualization](image2)

In the visualization technique by clusters we highlight the approach undertaken by Hilliges et al. [3] that addresses the issue of viewing and browsing music collections in the form of a ship radar (see Fig. 3). The solution is based on the similarity between music and four axes that represent characteristics of music: slow-fast, rough-clean, calm and turbulent-melodic-rhythmic. In the radar center point is the music selected by the user. The remaining tracks are positioned according to the similarity, that is, the closer to the center will meet more similar with respect to the selected music. To calculate the similarity between songs, as well as position the axle, the audio features are extracted.

![Fig. 3 AudioRadar visualization](image3)

2.2 Visualization and exploration of listening histories

Regarding the works from listening histories, we see that they differ in the way they represent the history viewing, which include a timeline visualization, graph and clusters.

Within the view of the historical timeline and chart we include the works of Dias et al. [4] [5] who intend to carry out the exploration of the history of a music library, and the user can infer their tastes and habits (see Fig. 4). The operation is done
through a timeline of cells and columns of dots. Each stack of points represents a day of music playback, each point represents a particular artist. The points are organized by size and color. The size of the dots represents the frequency of hearing a certain artist, and the color, how old is the music. The age of a music is measured by the difference between the year we are and the date of release of the album in question. Old songs are categorized with the color blue and the recent ones with red. The lines that connect various points indicate which of the same elements are present in the history.

**Fig. 4 Global visualization**

In the preview area history through clusters we highlight the work of Chen et al. [6] which proposes the solution HisFlocks that presents the playback history, focusing on the comparison and relation of genres and artists from various users. The history is presented to the user on a weekly timeline where artists are, mapped by similarity between them and clusters of genres (see Fig. 5).

**Fig. 5 Last.fm HisFlocks visualization**

The similarity of each artist is calculated based on the analysis of the semantic tags, which uses the Porter algorithm [7]. The mapping of each artist in the cluster that is most similar, is also calculated based on the similarity of tags. The representation of the tags on the timeline of artists, is given by the number of plays of songs of the same artist, that means, the greater the representation of the tag, the more number of reproductions have been played. To make it easier to compare historical user, an unambiguous color is used for each record.

2.3 Discussion

After performing an analysis of the most important techniques for visualization, we can identify the use of treemaps as the more suited to situations where we wish to view and explore the music collection in its entirety and without much detail. However, it is not as effective when you want to explore in detail the music collection, with a technique that does not favor innovation. The use of graphs as a technique for visualization and exploration of collections is applied to cases where it favors the visualization of similarity between songs, in addition to visualization and exploration of the music collection. This type of technique is simple and visually effective because the user can have a general perception of his collection. However, it can become confusing and tiring, if the collection has thousands of songs. The use of clusters, based on features extracted from the audio or metadata as a method of visualizing and exploring collections, produces diverse solutions, which can either be used for exploration of the collection or to perceive the similarity between the songs present in the collection.

Regarding visualization techniques for viewing the historical, timeline, is more common, and is easier to use when we want to display a history. This type of visualization allows us to explore listening histories with more or less detail. It is suitable for situations where the user wants to do a thorough search in his history. Visualization by clusters, like the graph visualization is more focused on solutions which aims to understand the relationship of similarity between songs previously heard.

In short, the visualization area and the exploring music collections presents more robustness in the way it presents the collection as well as the methods used to extract features of music in order to find ways of comparing these. The viewing area of the listening histories, try to give more information to the user about how we listen to the songs in our collection, and is more often concerned with the presentation to the user as simple and clear. Though neither of the two areas fills all requirements, together they can fill in the gaps that each in separate have.
3. Developed Solution

3.1 Objectives

In this work we intend to combine the visualization and exploration of a music collection with the listening history. Our main objective is related with the way users view and explore their collections. The solution is customized for each user, since the results will differ from user to user and even for the same user, it will differ for different hours since the display will take into account the listening histories. We want the user to rediscover his music collection through the history of reproduction, that is, we want the user, through his musical tastes for a specific time of day, could hear music appropriate for the time in question, which were already forgotten in his memory.

3.2 Extraction and organization module

Our solution is divided into two separate modules (see Fig. 6): extraction and organization of music collection and visualization.

The extraction and organization module is responsible for the work necessary for extracting the music collection, being also responsible for organizing the information to send the characteristics to the visualization module. This is divided into two stages: extraction of the music collection and organization of music collection.

In the extraction stage of the music collection we collect the required information using the community service Last.FM that extracts information from the history of music playback. Through this service we could collect for each song the following: song name, artist name, album name, album cover image, number of plays, date of last reproduction and list of similar artists.

In the organization stage of the music collection we organize the extracted information to obtain for each hour a set of songs heard more times and less time, considered as history songs, and a set of songs similar to the previous ones that have been listened less times and more time, considered recommendations songs.

First we perform a division of the collection per hour of play, where we put music played at the same hour on the same set. It is considered as some hour, the time between zero minutes and fifty-nine minutes of the same hour. After the division of the collection per hour, each set is ordered in decreasing order by F factor. The F factor is a ordering factor that uses part of the formula memory retention [8], which tells us how a song is forgotten in our memory, that means, songs heard less times and less often have a higher value and songs heard more times and recently have a lower value of F factor. With that we have calculated for each song the F factor, given by:

\[
F \text{ Factor} = \frac{\text{actual date} - \text{last replication date}}{\# \text{reproductions}}
\]

Equation. 1 - F Factor

In this way we obtained for each hour a set of music ordered from heard less times and less often, for songs heard more times and recently.

Fig. 7 Division of songs for each hour

Each pool was divided into two subsets (see Fig. 7), the first through the first third of elements in the set and the second subset with the last two-thirds. The first subset contains songs with higher F factor, referred to as set of recommended songs. In the second subset are the songs with minor F factor, called the set of history songs.
Fig. 8 - Division of the list of history music into subsets and comparison with list of recommended songs
Legend: 1 – List of history music
2 – List of potential recommended music
3 – Subsets of songs taken from the list of history songs
4 – List of recommendations for each subset

The set of songs from a different hour will be saved in a list for each time called list of songs from the history, while the set of recommended song will be stored in a common list to all hours, called list of potential songs to recommend.

Finally each list of songs from the history (1) is sub-divided into sets with the number of elements required for viewing (3) (see Fig. 8). The set of songs from the history would be allocated from the tail to the head, for consistency, being presented on display in the same order. For each slice of list of music from history (3) it was selected a part of list of potential songs to recommend (2). This selection took into account the similarity between artists of the songs present in the slice of music history, previously chosen, and the songs of the recommended list of potential songs. The songs that got equality between names of artists have integrated a new list named list of recommended songs (4). This list was ordered by the factor F of each song in a descending manner. In the end we obtained a list of recommended songs for each slice of the list of songs from the history.

We can thus conclude that for every hour we obtained a list of songs from the history, which when sliced gave a list of recommended songs for each slice.

3.3 Visualization module

The visualization module is composed by our solution interface. It is in this module that all the information from the server can be viewed. This information is presented to the user on the form of two sets of songs: recommended songs and history songs.

The interface of the solution is composed of 4 distinct areas (see Fig. 9): the area of the history songs (1), area of recommended songs (2), control area (3) and area of playlist (4).

As unit of representation of the music collection we chose a music, represented by the album cover to which it belongs.

Music

Fig. 9 BACH interface

Fig. 10 Buttons
Each song is represented by the album art that it belongs to and contains 4 buttons (see Fig. 10) button to add to playlist (1), button of music information (2), the similarity button (3) and preview button (4).

The playlist button located in the upper right corner allows users to add songs to the playlist until it is completely filled.
The information button located on the bottom right corner lets the user know more information about the song in question (see Fig. 11).

![Fig. 11 Music information](image)

The similarity button, located in the lower left corner, which indicates similar songs that gave rise to the music in question is found in this area of recommended songs (see Fig. 12). The similarity button has two different functions, if we move only the mouse over this button, it will perform a highlight, highlighting the recommended songs that are similar. If the user press the similarity in addition to the enhancement of similar music it can still browse these to get more information.

![Fig. 12 Similarity between artists](image)

The preview of the music button, located at the image of the album cover center, allows the user to hear a snippet of the music. This request for the preview of music is conducted using the echonest\(^2\) service, where we make a mark based on the name and artist of the song.

### History Area

![Fig. 13 History area](image)

The history area (see Fig. 13) is located on the external rectangle of the interface and contains 104 songs from the list of history songs. The songs are arranged by artists and are ordered alphabetically by artist name, from top to bottom, left to right, following the usual reading order. The songs are arranged by artists at the interface, since the calculation of similarity between songs is calculated according to the artist. Thus, all the songs that get similar F Factor, and are from the same artist are side by side.

### Recommendation Area

![Fig. 14 Recommendations area](image)

The recommendation area (see Fig. 14) is located at the interface inner rectangle, the center of the display, and contains 24 songs taken from the list of recommendations. As in the history area, the songs are represented by the image of the album cover to which they belong and are arranged alphabetically by artist name, from top to bottom, left to right.

### Control Area

![Fig. 15 Control area](image)

The control area of the display (see Fig. 15), located at the bottom of the interface consists of 24 buttons of hours (see Fig. 15 -1), and two sets of buttons to navigate on pages of list of recommended songs and list of history songs (see Fig. 15 -2)

The buttons of hours work by selecting with the cursor and allow the user to navigate through their music collection by viewing the music from the history area and recommended area for the selected time. Whenever the user presses an hour, the playlists from the history and recommended areas are discounted to the time in question. The selection of a different time of the current, causes the history list to be updated to the time in question which will consequently produce an updated list of the recommended songs.

Each set of buttons to browse through lists (list of songs from the history and list of recommended songs) consists of a rewind button and a button to advance. Between the buttons that make up each set.

\(^2\) http://the.echonest.com/
is the information from the page number they are viewing and total number of pages from the list. The set of buttons for navigating in each list is set according to the location of the command list, which means, the set of buttons that controls the list of recommended songs is towards this area, the same happens to the set of buttons that controls the list of songs from the history.

As mentioned in section organization of information the list of recommended songs is produced according to a slice of songs from the history. For this reason, whenever you browse the list of songs from the history, the list of recommended songs will be updated, which will lead to be viewed from the first page, being the values of the current page and page number of the list of recommended songs updated. The fact that the user selects a different time will require an update in lists, which will make each of them back to the first page and the total number of pages updated for each case. If the user selects the same time that is pressed, the history list will go back to the first page, which will lead to a list of recommended songs going back to the beginning, and again displaying the first 24 songs present in the list of recommended songs.

**Interaction client/server**

The exchange of information between server/client was carried out using Ajax requests. These requests allowed us to receive all the information necessary for the operation of our solution, since the playlists to the button images. From the requests made we highlighted the requests of both lists of music. It is at the start of the solution that more requests are exchanged for information between the display and the server. First we perform the highest priority order and will take longer to process, that is, lists of songs for the time in question. Then we perform the requests for images for the buttons that will be stored on the client side to reduce the number of requests to the server.

4. Experimental Evaluation

To test the solution we use two types of tests: a test where we define tasks for users, and a satisfaction survey.

4.1 Objectives

The test with tasks intended to test the functioning of the algorithm of organization of listening histories and music recommendation algorithm. To this end, we created a test scenario that is based on the simulation of the passage of time. This variable is crucial to test the proper operation of the solution, since the ordering and organization of music depend on this variable. Thus, we intend to recreate the normal use of the application. With this test we intend to comprehend if with the passage of time the mean value of the factor F decreases. The decrease of this value tells us that less heard songs have recently been heard, which indicates that the user heard recommended songs, as these possess the highest value F Factor.

With the satisfaction survey we intend to evaluate the interface as a whole and the features of the proposed solution. To evaluate the interface we inquire the users about the visualization, if this was readily apparent, or if there are shortcomings in the design of it. Regarding the features that are present in the solution interface, we evaluated whether the form of research and the similarity between songs were easy for the user perception.

4.2 Users

The tests were performed with users who were previously registered on the Last.fm service, which had a history of playing time for the test between 260 and 330 songs. Users were contacted via the Internet, being used the personal history of each user in the testings made.
Our group of users was composed of 11 people, all male. The age of our users was between 18 and 60 years.

Each user was asked to perform a set of tasks, which consisted in building playlists with the songs present in the various pages of the history and recommendation area for the indicated time. The time indicated differed from user to user, and was chosen for the hour that had the highest number of songs in the history area. The tests had no time limit, given that the execution time of each playlist would differ depending on the number of elements that the collection had, and how the person likes to explore his collection. After the tasks, the users were asked to answer to a satisfaction questionnaire, available in internet.

4.3 Results

To analyze the obtained results by tasks test we create a boxplot graph of the values of F Factor (see Fig.17), and a histogram showing the distribution of the values of F Factor (see Fig 18). We can see in the boxplot graph that the median value is gradually decreasing, which indicates that with the passage of time the average value of F Factor goes down as soon as songs that were not heard for some time or who were heard at other times were reproduced recently. We also acknowledge that the values of the quartile 2 are approaching zero, that means, there are more songs recently heard.

![Fig. 17 Boxplot](image)

With the histogram (see Fig. 18) we can see that in all ranges there are variations in the number of songs from playlist to playlist. This is due to the calculation of F Factor that are made for all new songs and to each new playlist, for the simulation of the passage of time. Each time the F Factor is updated for each song, and if it has not been heard, the value of F Factor will increase, so the number of elements in each interval will undergo changes, moving some of the songs for the next following interval. The most relevant for our analysis ranges are: the interval [0.0, 0.1] (recently listened songs) and [0.9, 1] (songs heard longer). In the interval [0.0, 0.1] we see that there is an increase of elements from playlist to playlist. This increase of elements indicates that more recently songs were heard that did not exist. This is due to two factors, the reproduction of songs that were not played for some time and/or reproduction of recommended songs (heard in another time). In the interval [0.9, 1] we found that there is a decrease in the number of elements between the playlist 1 and 5, which indicates that these songs were recently heard and passed to an interval with a lower F Factor value.

![Fig. 18 Music distribution by intervals of F Factor](image)

Through the responses of the satisfaction questionnaire we found that users understand the mechanisms present in the solution. In addition to understanding the solution, users liked to use it stating that they rediscovered their music collection.

5. Conclusions and Future Work

From the results obtained we can conclude that our solution meets the objectives we set ourselves. Users with the solution chosen to listen to music that wasn’t heard in some time, which led to the number of songs with the lowest F Factor to increase. The F Factor value calculated by the formula F factor contributed to our solution, in the way that, this was an important fact in the organization of information extracted from the music collection. This measure allowed us to organize and divide the music from collections into two groups: the history songs and recommended songs. These two groups have been central to the way we outlined the interface of the solution. We took the users to choose songs not heard for some time, leading them to rediscovering their collection, and we recommend songs that are not heard in a while but still are in accordance with the tastes of the users for a given hour.
Through the satisfaction survey we can conclude that the solution was easy to understand for users. The mechanisms from the interface, namely, the mechanism that allows the identification of the song or songs of the history that led to the recommendations, were understood and proved useful in exploring the music collection. The contribution made by innovating the way to visualize and explore music collection through the listening histories for each hour pleased the users. They wanted to continue to use the solution to visualize and explore their personal music collection, stating that they viewed and explored their collection from a point of view never used, rediscovering, through songs already forgotten, their music collection.

By analyzing the work we noted that several improvements can be made in future work. At the level of the solution interface we identified a need to integrate a fully functional player that allows full playback of songs presented in the solution interface. The interface should include, beyond the image of the album cover of the song, the song name in the form of text. The textual information has proved to be important, for the rapid identification by the users, of the songs present at the interface.

Given the lack of viability in the integration of the solution with the last.fm server, to extract album art, and the echonest, for previews of songs, there is a need to integrate the solution with one or more of other servers, to remedied the lack of information.

6. References


