Visualising the top songs of the twenty-tens

David Robín Karlsson

Abstract— A web app visualising the characteristics of the top hits released in the years 2010-2019 was developed. The clustering algorithm DBSCAN was utilised to find patterns in the songs, proving that most top hits sound the same. The web app was user-tested and the resulting data was used to formulate appropriate improvements suggestions. Suggestions for future developments are given as well.

Index Terms—Parallel coordinates, Musical trends, Clusters.

1 Introduction

At some point in life, everyone has thought to themselves ”the songs on the radio all sound the same, or am I going crazy?”. This is an interesting thought that is usually pushed back with responses blaming different kinds of biases, e.g ”you just don’t like the music”, ”you don’t listen closely enough” or ”you only listen to radio stations playing popular music”. What if there is a way to finally debunk this statement?

The aim of this project is to visualise the top songs and their characteristics in the years 2010-2019. A clustering algorithm and multiple visualisations are implemented to allow a quantitative analysis of the musical trends that defined the music of the past decade. With this, the question can be answered, do all hit songs all sound the same?

2 Background and Related Work

A simple analysis of the top songs of 2019 using Python was made in 2019 [4]. This analysis uses, in practice, a subset of the dataset used in this project and visualises only a few parameters using bar charts. This project differs mainly in that almost all parameters in the dataset are visualised with techniques that allow for analysis of parameter correlations and clusters. This project also uses data covering several years and provides interactivity to the user, not just static images.

3 Data

The dataset [2] is compiled by Spotify and was extracted using their API. Spotify calculates a number of useful features for all songs in their internal database, which formed the basis of this project. The dataset contains 603 entries of songs with the following parameters:

- ID - A unique ID for a song in the data set.
- Title - Title of a song.
- Artist - Artist of a song.
- Top genre - Spotify’s best fitting genre of a song.
- Year - Release year of a song.
- Tempo - The tempo of a song measured in bpm.
- Energy - How energetic a track is perceived, values are in the range $[0, 100]$ where 0 is the lowest amount of energy.
- Danceability - How suitable a song is for dancing, values are in the range $[0, 100]$ where 0 is the lowest amount of danceability.
- Loudness - The overall loudness of the track measured in decibels.
- Liveness - A probability measure indicating the chance that a song is performed live. Values are in the range $[0, 100]$ where 0 is the lowest chance of a song being performed live.
- Valence - A measure of how happy a track is, values are in the range $[0, 100]$ where 100 is the highest amount of happiness.
- Duration - The length of a song measured in seconds.
- Acousticness - A probability measure indicating the chance that a song is acoustic. Values are in the range $[0, 100]$ where 0 is the lowest chance of a song being acoustic.
- Speechiness - A measure of how much a song is dominated by spoken words. Values are in the range $[0, 100]$ where 0 represents completely acoustic songs.[1]
- Popularity - The popularity of a track ranked by an algorithm focusing mostly on the number of plays. Values are in the range $[0, 100]$ where 0 represents very unpopular songs.

The ID parameter was not visualised, it was only used in algorithms. The Top Genre parameter was not used as the values were too specific to be recognisable to a layman. Since everybody listens to music with different volume, the parameter Loudness was not used as it was deemed too subjective compared to the rest of the parameters. The dataset is stored in a comma separated values file (.csv).

4 Method

The web app was designed with the *Visual Information–Seeking Mantra* [5] of overview → filtering → details-on-demand in mind. To fulfil this mantra, three different visualisations were used: a timeline showing how the song features have developed over the years, parallel coordinates representing all songs in the dataset and a radar-plot to visualise the parameters of a single song.

The timeline provides an overview of the data through the average of each year’s song features, visualised within the timeline using a multi-line plot with an accompanying legend right beside it. Filtering is done through *brushing* to select which years’ worth of tracks will be visualised in the parallel coordinates. This covers overview and filtering. Since the release year of a track is a discreet value (e.g no specific dates, only year) and brushing implies continuous data, a text element showing which years are selected was added to ease the filtering.

The parallel coordinates provide an overview of all songs and their features. Filtering is done through brushing on any axis to select songs with features within a certain range. When hovered above, any line in the parallel coordinates will display the song name and artist. If a line is clicked, that particular song’s features will be visualised in a the radar plot. With these features, the parallel coordinates covers

• *David Robín Karlsson is with Linköping University, Sweden, e-mail: davka030@student.liu.se.*

[1] Most songs have a speechiness in the range $[1, 25]$ because Spotify ranks songs containing both speech and instruments low. Only podcasts and speech-heavy genres like rap rank higher.

overview, filtering and details-on-demand. To allow analysis of correlations between all pairs of parameters, functionality allowing free placement of each axis was implemented. Since searching for a particular song can be hard by looking at the lines alone, a separate list showing all song names currently visible in the parallel coordinates was implemented. This list also exhibits the same hovering and clicking behaviour as described earlier. To ease brushing, a button with the text ”clear brushes”, with the described functionality, was implemented to the side of the parallel coordinates. Some additional info explaining parallel coordinates and a legend was implemented as well.

The radar plot visualise the details in details-on-demand. The radar plot visualise the song features Danceability, Energy, Speechiness, Acousticness, Valence and Liveness. The parameters song name, artist, release year and tempo are visualised separately with text since these attributes do not have compatible value ranges as the aforementioned ones, which is essential in a radar plot. When no song is selected, the text ”Please select a song below :)” will be displayed to encourage selection.

The footer of the page contains credits to the creators, a link to the course web page and a link to the source code.

The implemented clustering algorithm is DBSCAN [1]. The resulting clusters are visualised in the parallel coordinates using differently coloured lines. The parallel coordinates legend explains which colours represent clusters and which colour represents noise. The algorithm uses the numerical song features Tempo, Energy, Danceability, Liveness, Valence, Acousticness and Popularity.

5 Implementation

The web app was developed with HTML/CSS and JavaScript. The JavaScript library D3 was used for loading the dataset and creating the visualisations since this is what D3 is specialises in. Development was done on a local Node.js server. Since this project focused on the visualisations and not necessarily website design itself, no consideration was taken to responsiveness. As a consequence, the website works best for a large and wide monitors. The colour scheme used was chosen to mimic Spotify's colour profile in an attempt to give the web app an official look, as if Spotify created it.

6 Results

The results of the project will be presented in this chapter. As seen figure 1 and 4 the dataset only contains a single cluster. This was true for all non-trivial parameters in the DBSCAN clustering algorithm, only the amount of songs in the cluster varied. With the final parameter configuration (ε = 38, minPts = 20), 92% of all songs fit within the cluster.

An issue with this design is the reliance on the user's experience with information visualisation applications. No explicit indication is given on how to brush on the parallel coordinate axes. There is also an issue stemming from the parameters themselves, no explanation for what the unconventional features liveness or speechiness is given.

When running on the developers computers no kinds of latency or choppy animations are present. This is most likely due to the fact that the dataset is relatively small ($n = 603$).

7 Evaluation

In order to test the user-friendliness of the web app, a usability evaluation was conducted. The test was designed to give insight into learnability, efficiency and eventual user errors, inspired by Nielsen [3], for a new user.

All users are introduced to the test by a prepared script, in which an explanation of the web app's visualisations is given as well as four tasks that needs to be completed. The user will not see the web app before the test begins to ensure that learnability can be observed and measured. The four tasks were:

- Task 1: Which years did the song have the highest respectively the lowest valence?
  (Focusing on the information in the timeline)

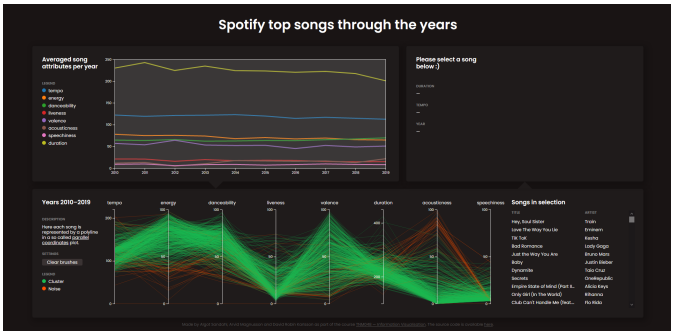

Fig. 1. The web app as it appears at first visit.

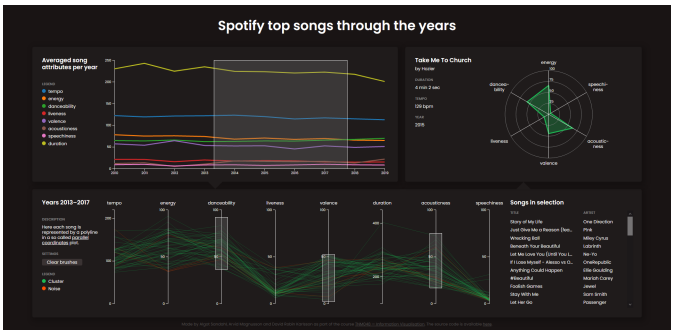

Fig. 2. An example of filtering in the timeline, filtering in the parallel coordinates and selection through details-on-demand in the radar plot.

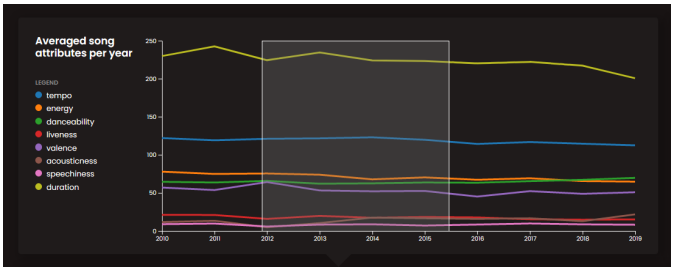

Fig. 3. A closer look at the timeline and filtering.

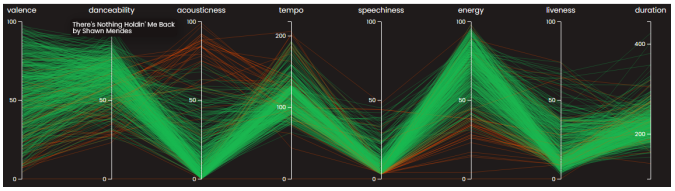

Fig. 4. A closer look at the parallel coordinates and the cluster after moving multiple axes.

Fig. 5. A closer look at the radar plot.

Table 1. Time consumed to solve each task.

Tasks	User 1	User 2	User 3
Task 1	17 sec	32 sec	20 sec
Task 2	1 min 50 sec	1 min 50 sec	47 sec
Task 3	1 min 34 sec	2 min 40 sec	2 min 45 sec
Task 4	2 min 24 sec	45 sec	20 sec

- Task 2: What is the pattern for most top songs?
  (Focusing on the clustering in the parallel coordinates)
- Task 3: Is it possible to identify outliers and filter out all songs but these?
  (Focusing on clustering and filtering in the parallel coordinates)
- Task 4: Find which song had the highest energy in 2013 and how long it is.
  (Focusing on filtering in the timeline and selection in the parallel coordinates or radar plot)

The completion of each individual task is timed to give quantitative data regarding the web app's learnability and efficiency. Any error the user makes will be noted by the test-giver. The user is informed that the test-givers wont aid in solving the tasks but can repeat what the different visualisations represent.

After the tests are completed a verbal questionnaire is completed in which the test-givers ask the user about different aspects of the web app. To give quantitative data, the majority of the questions are answered with a scale from 1 to 10 where 1 represents something very negative and 10 the opposite. The questionnaire is ended with open ended questions without adhering to any numerical scale to give the user the ability to freely comment about the web app and provide suggestions for improvements.

Three different user tests and questionnaires were conducted and the results are presented in tables 1, 2 and section 7.1, 7.2.

7.1 Open ended questions

Was there any interaction you thought was unclear or hard to understand?

- User 1: Filtering was hard since there was no hint that it was possible.
- User 2: That brushes exists. What the vertical axis in the timeline means. The difference between the red and green colour in the parallel coordinates, red could be interpreted that the songs are bad somehow.
- User 3: Unclear that filtering was possible in the parallel coordinates. Interactions without any hints to their existence was hard to find.

What do you think it means when some songs are coloured red in the parallel coordinates?

- User 1: Does not understand.
- User 2: That they do not fit into the general musical trends.
- User 3: That they to not fit the pattern that most other tracks do.

Suggestions for improvement?

- User 1: A hint on how to brush.
- User 2: Indicate that you can brush on the timeline and parallel coordinates. Change the colour of noise in the parallel coordinates. Implemented a button that only shows the noise in the parallel coordinates.
- User 3: Some kind of hint to brushing.

7.2 Test-giver notes and observations

Although some tasks took some experimenting to complete, no user was unable to complete any task. The hardest part for all testers was realising that filtering is possible in the timeline and the parallel coordinates. Once that hurdle was overcome the tasks involving filtering were quickly completed.

8 Conclusions and Future Work

A web app visualising characteristics of the last decade's top hits, following the Visual Information–Seeking Mantra, was successfully developed as presented in section 6. The implemented clustering algorithm showed that 92% of all hit songs are similar to each other, proving that most songs on the radio really do sound the same. The web app can be used in a multitude of other ways, e.g studying correlations between song parameters, finding interesting outliers year-by-year or searching for new songs you might enjoy using your preferred features. This is largely owed to the extensive filtering and selection capabilities.

As evident in the evaluation presented in section 7 there are some issues with the web app. The tasks and subsequent questionnaire proved that the most obvious flaw is the lack of hints towards brushing functionality on both the timeline and the parallel coordinates as all testers faced this issue in task 3 and 4. This issue was reinforced further by the answers to the open ended questions. This would be a top priority to improve.

Another issue that was touched upon was the cluster colouring in the parallel coordinates. User 2 indicated that the red colour to mark noise in the dataset could be interpreted as bad songs. The fact that the parallel coordinates legend labels red as noise with no explanation that it means noise compared to the cluster, not audible noise, could reinforce this misconception. This could be remedied by switching the two cluster colours to something contrasting without any inherit conventional meaning (e.g red = bad). Combining this with a text switch to rename noise to something more descriptive like "outliers" will improve clarity of the parallel coordinates.

Barring all mentioned fixes, there are two main paths forward to extend the functionality of the web app. The most natural improvement would be to integrate Spotify's app by giving the user the option to open a selected song or artist's page in Spotify. The other improvement would be to include more years worth of data into the dataset. This can be implemented elegantly with access to Spotify's API where data can be retrieved at will.

References

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[2] Leonardo Henrique. Top spotify songs from 2010-2019 - by year, 2019.

[3] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994.

Table 2. Questionnaire results.

Questions	User 1	User 2	User 3
How good/bad was the overall impression of the web app?	10	8	8
How good/bad was the colour scheme of the web app?	10	9	10
How easy/hard was it to complete the tasks?	8	6	6
How well telegraphed were the interactions between the visualisations?	10	7	4
How easy/hard was it to understand the information in the timeline?	7	5	10
How easy/hard was it to filter out data using the timeline brush?	8	5	8
How easy/hard was it to understand the information in the parallel coordinates?	10	9	9
How easy/hard was it to filter out data using the parallel coordinates brushes?	7	5	5
How easy/hard was it to understand the information in the radar plot?	NaN	10	10
How easy/hard was it to select songs to visualise in the radar plot?	NaN	10	10

[4] S. Pierre. Analysis of top 50 spotify songs using python. *towards data science*, 2019.

[5] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.