Preliminary analysis

Table of Contents

[Scope of Work 2](#_Toc416729489)

[Problem Statement 2](#_Toc416729490)

[Preliminary data analysis 3](#_Toc416729491)

[Data availability 3](#_Toc416729492)

[Data distribution 3](#_Toc416729493)

[Proposal 4](#_Toc416729494)

[Time estimation 5](#_Toc416729495)

[References 6](#_Toc416729496)

# Scope of Work

The Million Song Dataset challenge is a Kaggle competition hosted in 2012 which required to propose and implement a song recommender system able to predict which song a user is most likely to listen to, based of his listening history and the listening history of other users.

The scope of this case study, based on the interpretation of your request, is to propose, implement and evaluate the performance of a model:

* Which has not been studied before
* And compares well with the top models of the mentioned competition

# Problem Statement

Given:

1. The full listening history for 1M users
2. half of the listening history for 110K users (called the *visible listening history*)

Propose a ranked list of 500 songs per user, in decreasing probability of this particular user listening to that song.

The results will be evaluated using the Mean Average Precision (truncated at 500) of the proposed ranking against the second half of the 110K users listening history.

This second half of the listening history was not available to the competitors, which needed to send their proposition and receive in return the result of the Mean Average Precision through a web interface hosted at Kaggle.

# Preliminary data analysis

## Data availability

The Million Song Dataset Challenge has been closed in August 2012, but its datasets are still available.

Retrievable data:

1. The full Million Song Dataset
2. the *visible listening history* from a Kaggle GitHub repository, in the form of a list of 1.450.933 triplets of type <user>, <song>, <listening count>

It appears that the second half of the listening history is not available, and the access to the web interface to evaluate the performance of a model can’t be access as the competition is now closed.

In absence of this second half and the evaluation web interface, it is important to create a new training and validation set and implement the validation algorithm.

This can be done either by:

1. Splitting the *visible listening history* again by 2 and define one half as the validation set.
2. Building a new training and validation set from the full Million Song Dataset

## Data distribution

After a high level look at the *visible listening history*, the following was observed:

1. Approximately half of the users (~49%) listened to between 5 and 10 songs, which means their full listening history would show between 10 and 20 songs
2. More than half (~58%) of the listening triplets of the *visible listening history* have a count of 1, meaning this particular user listened to that particular song only one time
3. On the 163,206 songs present in the *visible listening history*, approximately 40% have been listened by only 1 user, and 85% only by up to 10 users out of the 110K users (0.01% of the users), meaning most of the songs are not very popular. Only 0.34% of the songs have been listened by more than 100 user, with some popular songs being listened by up to 5043 users (~5% of the users).

# Proposal

A good start would be to reproduce the solution of the winner of the competition [1], using a combination of user-based and song-based collaborative filtering and a configurable scoring function to obtain good results.

In this solution, each triplets are considered as a binary indicator saying whether a user have listened to a song or not, meaning simplifying the feature construction to concentrate more on the similarity measure and scoring function improvement.

Given the 2 last observations in the Data Distribution section, it seems that:

* The listening triplets wouldn’t contribute equally to the result of a ranking algorithm, as the biggest part is constituted by a user listening one and only one time to a particular song (observation B). Intuitively, that information should not be considered as important as when a user listens multiple times to a certain song.
* The songs vary greatly in popularity (observation C), leading to a popularity bias (which is a common problem in recommendation systems [2]).

Therefore, I propose to investigate the possibilities to mitigate those 2 problems. There could be several ways to address those, including:

* Using a more complex feature construction method, maybe a normalized model like TF-IDF used in Text Categorization. Using such a feature construction method might not perform well with the cosine similarity so, depending on the data distribution, we might need to change the similarity measure as well
* Using the cosine similarity with a binary indicator, but find a scoring function ranking the songs higher if users listen more to them, and at the same time lower if the song is popular
* Using content-based models to find similarities between songs and use it to lower the impact of observation B by grouping songs together. This would require using additional data that could be found in the full Million Song Dataset, like lyrics, genre, beat information, etc.
* …

Each of those proposals would require to go through the current literature to understand what the state-of-the-art is and propose a new solution accordingly

# Time estimation

1. **Setting up the training and validation set, and performance evaluation**

The first task would be to recreate a usable training and validation set, implement the evaluation algorithm, and compare the numbers to those given in the Kaggle tutorial for the baseline algorithm (giving the most popular songs to each users, filtering out the songs that the user already listened to)

Given that in the actual *visible listening history*, 49% of users listened to between 5 and 10 songs, splitting this dataset again would mean that 49% of users listened to between 3 and 5 songs, which might be a poor dataset for training.

**Conclusion:** Unless there is a way to recover the second half of the listening history, a new training and validation set would need to be built from the full Million Song Dataset in order to feed usable data to the ranking algorithm.

1. **Implementing the current winning solution**
2. **Understanding the current state of the literature on the proposed solutions, and propose a new one if needed**

Based on the time frame relative to a job interview, I would propose to investigate the recent developments of scoring functions to find one capable of balancing the effect of the popularity bias and “poor user history”, and propose a new one if needed. Such a scoring function should allow optimization using one or more parameter(s) to control the impact of each aforementioned problems

1. **Implementing the scoring function and evaluate its performance across different value of its parameter**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Man days**  **(days)** | **After work**  **(weeks)** |
| 1 | Setting up the training and validation set, and performance evaluation | 4 | 3 |
| 2 | Implementing the current winning solution | 5 | 3 |
| 3 | Searching the literature for a suitable scoring function, and propose a new one if needed | 10 | 7 |
| 4 | Implementing the scoring function and evaluate its performance across different value of its parameter | 5 | 3 |

So a total of **24 days full time**, or **4 months after working hours**.

# References

[1] Fabio Aiolli (2012): A preliminary study on a recommender system for the million songs dataset challenge

[2] Yading Song, Simon Dixon, and Marcus Pearce (2012): A Survey of Music Recommendation Systems and Future Perspectives