

Music-Map: A Music Recommendation System Based On Data Mining Technique

Basundhara Dey

ESL Lab, Dumdum, Kolkata

Abstract: *This report describes the design, implementation and protocol of a Music Recommendation System in details. I have used python programming language to develop the proposed system and SQLite to design and control the backend database. The report contains algorithm and data structure as well. In this paper, I present a music recommendation system, which provides a personalized service of music recommendation.*

I. Introduction

With the growth of the World Wide Web, a large amount of music data is available on the Internet. In addition to searching expected music objects for users, it becomes necessary to develop a recommendation service. I use Python to build this as it helps to build most lightweight platform and offers variety of functionalities. During the design of this project our main aim was to make the system easy accessible.

Music Map will consist of the following functionalities in brief:

- Recommendation on basis of age group.
- Recommendation on basis of region group.
- Recommendation on basis of language difference.
- Store your most played songs as your favourite list.
- Play the music online.

For each music object, the representative track is first determined, and then four features (Genre, Artist, Language and Duration) are extracted from this track. According to the features, the music objects are properly grouped. For users, the personal database table and history are analyzed to derive user interests. The content-based, collaborative and statistics-based recommendation methods are proposed, which are based on the favourite degrees of the users to the music groups. A series of experiments are carried out to show that this approach is feasible.

II. Motivation

With the explosion of network in the past decades, internet has become the major source of retrieving multimedia information such as video, books and music etc. Concerning a large amount of various data available on the Internet, there are existing websites which provides services for users to look for useful data. For text data in WebPages, the websites providing keyword-based searching or recommendation are developed, such as the search engine of Yahoo! [Yahoo] and the book recommendations of Amazon [Amazon]. For multimedia data, however, the websites providing such kinds of services are still limited.

People have considered that music is an important aspect of their lives and they listen to music, an activity they engaged in frequently. Regarding the music recommendation, a preliminary Music, as a powerful communication and self-expression approach, therefore, has appealed a wealth of research.

Advantages of Music Map

Music recommender is there to help users to filter and discover songs according to their taste, age, region and language. Meanwhile, the development of recommender system provides a great opportunity for industry to aggregate the users who are interested in music.

Standard retrieval systems force users to discover their favourite musical pieces by using queries including song's title or artist names. To overcome this limitation, many researches focus on music information retrieval (MIR), which enables users to discover their favourite pieces on basis of musical content such as genre, rhythm and melody. However, many users have difficulty in stating their musical preferences as queries. In fact, most users of music streaming services that let one freely listen to numerous songs for a flat fee, want to access their favourites one after another without querying. Recommender systems should thus be able to select musical pieces that will likely be preferred by estimating user preferences. So far, two major techniques have been proposed in this Music-map: user-data based and user-history based.

This system is here to stay for long as music is a universal, world-wide standard. Therefore, you can count on it being around for years to come. As more people begin to use recommendation technology, the features will automatically be evolved with time.

What Is Python?

Python is an interpreter, high-level programming language, pure object-oriented and powerful server-side scripting language for the Web. Like all scripting languages, Python code resembles pseudo code. It's syntax's rules and elegant design make it readable even among multi-programmer development teams. The language doesn't provide a rich syntax, which is really helpful. The idea behind that is to keep you thinking about the business rules of your application and not to spend time trying to figure out what command you should use. It is also true that Python is interactive, portable, easy to learn, easy to use, and a serious language. Furthermore, it provides dynamic semantics and rapid prototyping capabilities.

WHY PYTHON?

1. Python is simple

Python has clear and simple rules, and is closer to English than any of the languages I use. Creating Python programming is so straightforward that it's been called "programming at the speed of thought." Statements are terminated by end of line, and block structure is indicated by indentation. Python programs look like executable pseudo-code. This eliminates a host of troublesome errors for beginning programmers, especially placement of semi-colons, bracketing and indentation.

The simplicity of Python makes it easy to learn. In addition to the list (dynamic array) data structure, Python provides tuples (immutable lists) and dictionaries (hash tables). Together with the class mechanism, these can be used to quickly build sophisticated data structures for interesting projects. The absence of type declarations makes for less code and more flexible programming.

2. Python is self adjustable

Python provides full dynamic run-time type checking and bounds checking on array subscripts. Python employs garbage collection so there is no problem with dangling pointers or memory leaks. It is impossible for user code in Python to produce a segmentation violation. In this respect Python is similar to Java, and both are much safer than C++

3. Python supports object-oriented programming

Languages like C#, Java, and Python are all object-oriented. But Python does them one better. In C# and Java, OOP is not optional. This makes short programs unnecessarily complex, and it requires a bunch of explanation before a new programmer can do anything significant. Python takes a different approach. In Python, using OOP techniques is optional. You have all of OOP's power at your disposal, but you can use it when you need it. Got a short program that doesn't really require OOP? No problem. Got a large project with a team of programmers that demands OOP? That'll work too. Python gives you power and flexibility.

4. Python Runs Everywhere

Python programs are *platform independent*, which means that regardless of the operating system you use to create your program, it'll run on any other computer with Python. So if you write a game on your PC, you can e-mail a copy to your friend who runs Linux or to your aunt who has a Mac and the program will work.

5. Python Is Open Source

Python is free. You can install it on your computer and never pay a penny. But Python's license lets you do much more than that. You can copy or modify Python. You can even resell Python if you want (but don't quit your day job just yet). Embracing open-source ideals like this is part of what makes Python so popular and successful.

Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Summary

- easiest interface to handle
- highly scalable, suitable for large projects as well as small ones
- rapid development
- portable, cross-platform
- embeddable
- easily extensible
- object-oriented
- simple yet elegant
- stable and mature

Why This Project?

- Create, store and manipulate different data structures.
- Relays current and updated information needed for accomplishing recommendation.
- Reduces the effort of human in data entry like genre, artist, new released etc.
- Ensures and easy, impartial, and orderly approach in addressing specific needs without any confusion.
- Reduces operation time and searching time significantly.
- Consequently it allows man to be relaxed in better way.
- Cost-effective
- The exchange of stored and manipulated information between more than one user or participants possible.

Recommendation does not always mean that the site will do everything from start to finish by clicking a button. User can also play his favorite list or update and modify the list. But to give him friendlier environment, it will make logical, algorithmic decisions to make the suggestion easier and more efficient. Thus automation reduces human effort by introducing it in relative arena.

Recommendation system based on Data Mining

In the proposed system I have achieved the following functionalities:

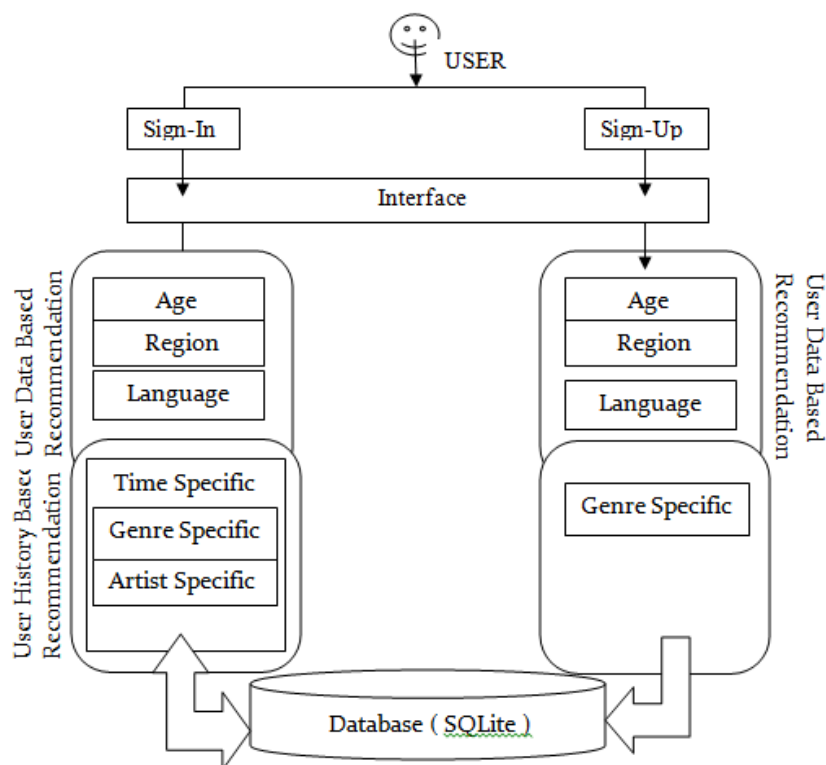
User-data based recommendation: While the user signing-up in the site, it will ask for various information like age, region, language etc. Upon submitting all the relevant information, the system will search in the database for that same age group, same region and same language. A mid age person's choice certainly differs from a teen age. Difference also comes with language and region. On basis of this logic, a user's age will be tallied with his same age group and thus the recommendation comes in his recommendation page. This same logic follows for the other parameters too. It also recommends the user by considering how someone else rated other songs. For example, suppose that there is a target user who likes piece A. If many others like A and B, B will be recommended to the user. This technique is widely utilized in practical e-commerce services (e.g., Amazon.com and the iTunes music store) and has been demonstrated to be rather effective.

However, there are two problems. The first problem is that pieces that have not been rated (e.g., newly released CDs and less well-known songs) cannot be recommended. This is known as the new-item problem or the cold-start problem. Therefore, the chances of encountering unexpected favourites are limited. The second problem is that the artists of the recommended pieces tend to be the same and are often well known because most users tend to rate highly musical pieces by the same artists. Such recommendations are unsatisfactory or meaningless.

User-history based recommendation: For a regular user, our database will be enough enriched with his playlist items. While saving his play history, the backend database filters them according to genre and artist as well as time. Thus after a while, the system will have enough data to know what is the choice of the user. If in the evening, any particular user is fond of listening A kind of song, then while the user sign in at the evening time later (of course it have a time group), the recommendation suggests it some more A kind of songs to select. Moreover, this method recommends musical pieces similar to the users' favourites in terms of musical properties. This results in a large variety of artists; i.e., various pieces are recommended even when they have not been rated. However, these methods have essential problems concerning accuracy of recommendations because similarity in content is only one of many factors characterizing user preferences. In addition, it is difficult to associate user preferences with musical content by using a real database where most users provide few rating scores. Unfortunately, reliable methods of doing this have not been established.

III. Methodology

BLOCK DIAGRAM



PROTOCOL

A successful music recommender needs to meet users' various requirements. For user-oriented design, lots of efforts on user studies need to be investigated.

User data collect, as the one of the key elements; it models the difference in profile. For example, the difference in geographic region or age or language, their music preferences might be different. Interestingly, other factors such as gender and interests could also determine their choices of music. It is therefore essential in prediction of their music taste.

It has been divided into two parts: user profile and user experience.

First Step - User Profile. It suggests that the user profile can be categorised into three domains: demographic, geographic, and psychographic

Demographic	Age, marital status, gender etc.
Geographic	Location, city, region, Language etc.
Psychographic	Interests, lifestyle, personality etc.

Here for this system, I have used three parameters: Age from Demographic; Region and Language for Geographic. When a user first time log into the system, the system only reflects the recommendation based on the above parameters. The comparison will be done with the other database of several users who have accounts in the system already.

Second Step - User Listening Experience. From the Psychographic row, here Interest has been chosen as one of the parameter in this system. When regular users sign-in the system, the database automatically saves their interests for every single logging in. The system looks for the day-time while he is listening the songs as well as keeps track of the genre and artists.

So for next time onwards, the system will add new features along with the age, region and language. It will show recommendation on genre and artist based on time. For example, user X has listened Jazz in most the evening hours. So upon signing in at evening, system will recommend Jazz songs in the genre recommendation. Same applies to artist as well.

Another issue is the sequence of the favorite playlist. Though the most similar songs are given in order, the theme and mood can be dramatically changed in between. This may result in the dissatisfaction and discontinuation of the songs. Rather than randomly shuffling, human skipping behavior can be considered for

dynamic playlist generation. With the help of skip counter and play counter, system can detect the most played and list heard piece.

Database

➤ *For User Entry*

Column ID	Name	Type	Not Null	Default value	Primary Key
0	fname	varchar(20)	0	null	0
1	lname	varchar(20)	0	null	0
2	age	int	0	null	0
3	country	str	0	null	0
4	language	str	0	null	0
5	sex	char(1)	0	null	0
6	id	varchar(20)	0	null	0
7	password	varchar(20)	0	null	0
8	cnf_passowrd	varchar(20)	0	null	0

➤ *For Song Entry*

Column ID	Name	Type	Not Null	Default value	Primary Key
0	s_id	varchar(20)	0	null	0
1	name	varchar(20)	0	null	0
2	genre	varchar(20)	0	null	0
3	artist	varchar(20)	0	null	0
4	language	varchar(20)	0	null	0
5	duration	float	0	null	0

➤ *For User Preference*

Column ID	Name	Type	Not Null	Default value	Primary Key
0	id	varchar(20)	0	null	0
1	s_id	varchar(20)	0	null	0
2	time	float	0	null	0
3	genre	varchar(20)	0	null	0
4	artist	varchar(20)	0	null	0
5	play_count	int	0	null	0
6	skip_count	int	0	null	0

IV. Conclusion

This paper presented a basic metadata-based model and two popular music recommender approaches: User profile and user listening experience. Though they have achieved great success, their drawbacks such as human efforts are obvious. However, this research is still at an early stage.

As I can see from the development of music recommenders over the past years, the given results tend to be more personalized and subjective. Only considering the music itself and human ratings are no longer sufficient. A great amount of work in recent years have been done in music perception, psychology, neuroscience and sport which study the relationship between music and the impact of human behavior. Undoubtedly, music always has been an important component of our life, and now I have greater access to it. All of these highlight that music recommender is not only a tool for relaxing, but also acts as an effective tool to meet our needs under different contexts. To our knowledge, there is few research based on these empirical results.

Designing a personalized music recommender is complicated, and it is challenging to thoroughly understand the users' needs and meet their requirements. As discussed above, the future research direction will be mainly focused on user centric music recommender systems. A survey among athletes showed practitioners in sport and exercise environments tend to select music in a rather arbitrary manner without full consideration of its motivational characteristics. Therefore, future music recommender should be able to lead the users reasonably choose music. To the end, I am hoping that through this study I can build the bridge among isolated research in all the other disciplines.

V. Future Work

Security features need to be imported within this system in order to prevent vulnerabilities and secure the information from the outside attackers. Further work might explore the exceptions that could hinder the proper functioning of the system and once detected can be easily removed by some simple changes in the program code.

An additional algorithm has also been developed to detect the mood of the user. On basis of the first song, the user listen, system will search the next similar song. Thus it will be easy to recommend the most appropriate song on basis of user's mood.

I hope that the implementation details and the concept of this project will aid the wireless technology community. With some due work this system could become practically implacable and a huge success.