

# Creating a Hybrid Music Recommendation System from Content and Social-Based Algorithms

Jamie Cai, John Francis, Stephen Gheysens

*mirkwoodheiress@gmail.com, johnfrancis12@msn.com, stephengheysens@yahoo.com*

Rutgers University, GSET '09

## Abstract

Currently, most music recommendation systems are either social or content-based. However, both of the existing systems have individual advantages and disadvantages. The goal of this paper was to rectify this problem by creating a new system that would combine both social and content-based music recommendation algorithms, also known as hybrid music recommendation. The output of social recommendations gathered from the music recommendation sites Last.fm and Pandora was used to compare audio features of music that were extracted using MARSYAS. These values were then inputted into a Support Vector Machine in order to "train" it to be able to compare content of songs. The results gathered indicate that the system created is capable of recommending songs effectively, though not as well as music recommendation sites such as Last.fm and Pandora.

## Introduction

Current online services, such as Last.fm and Pandora, attempt to recommend music to a user based on the user's interests. Up to this point, most music recommendation sites have used one of two approaches for recommendations: social-based or content-based, both of which have specific advantages and disadvantages.

Social-based recommendations suffer from a few issues. If a song has not yet been rated by a user, it cannot be recommended. Also, social-based recommendations depend on an adequate number of ratings by a user for the system to accurately assess the user's interests, so even if a track is supposed to be similar to a user's taste, the user may not always enjoy the proposed music [1]. Content-based recommendations eliminate the aforementioned problems, but the process of analyzing only music data to make recommendations has yet to reach the accuracy of current social-based methods [2].

Content-based recommendations also assume that the user will know what music he/she wants to search for. Therefore, users may never find music they have not heard of before. This paper centers around the possibility that combining these two methods into one algorithm will improve the recommendation results because it will have the advantages of both methods. A similar method was proposed that runs separate social-based and content-based algorithms and combines the results [7]. This paper is intended to determine if a single algorithm is a better approach.

The system detailed in this paper is based around an initial music library of about 1,990 songs. For the social-based aspect, a list of similar songs for each song in the library was retrieved from Last.fm and Pandora. These sites were chosen because of their substantial user and music bases. They also differ in the way they recommend music; Last.fm is a social-based site while Pandora is a music analyst-based site. To run the content-based aspect, features from the previously determined similar songs within the library were extracted and entered into a Support Vector Machine (SVM) for training. Next, the features of a song chosen by the user were extracted, compared to features of the other songs in the library, and analyzed by the SVM to output a list of recommended songs within the music library [3].

A completely implemented system would need a library with millions of songs. The experiment in this paper assumes that a sufficient number of songs are present in the library to train the SVM for the algorithm.

## Background

This project focused mainly on audio engineering, which was used in order to gather and classify music based on content.

## **Music Information Retrieval**

Music recommendation is based on the idea of music information retrieval. There are three main types of music information retrieval: content-based, social-based, and hybrid (a combination of the aforementioned).

### *Content-Based Recommendation*

In general, content-based music recommendation uses the audio features within a track, such as melody, harmony, bass, tempo, etc. After that information is retrieved, there are different ways of organizing the results. Methods range from assigning encoded information to music that can then be searched for (known as metadata), organizing music based on musical concepts (rhythm, melody, pitch, etc.) [4], having editable user profiles that are created based on a user's liking or disliking of various songs [2], and even comparing the intensity of audio signals [4].

### *Social-Based Recommendation*

Social-based music recommendation, also known as collaborative filtering, has its foundation on users' ratings and recommendations of songs. There are three main methods of social-based music recommendation: the content-based method, which recommends music based on the songs that a user listens to, the collaborative method, which recommends music based on groups a user is part of, and the statistical method, which recommends music based on statistics [1]. Social recommendation has an advantage over content recommendation because users are able to find different types of music since they are linked to songs based on a combination of their own and other people's likes and dislikes.

### *Hybrid Recommendation*

In order to try and fix the problems with music recommendation, there have been attempts to combine content and social based music information retrieval. One example of this is hybrid music recommendation, which is an extremely advantageous option because it blends together content and social data. There are two main ways that hybrid recommendation works: combining the individual outputs of content and social based recommendations or integrating them into a single

algorithm [7]. For the purposes of the paper, the latter option was used in a distinctive format by applying the use of an audio feature extractor and a support vector machine.

## **Software**

Different types of programs were used within this project in order to successfully gather information about music tracks and classify them as being either similar or dissimilar.

### *MARSYAS*

In order to retrieve the information needed for our approach to music recommendation, we used the open source C++ library known as Music Analysis Retrieval and Synthesis for Audio Signals (MARSYAS). This library includes a powerful feature extractor known as bextract. Bextract gathers 34 different audio features from within a music track and classifies them with a value. The values are then written to a text file [5]. Every line that is written is called a feature vector, which can then be used by machine learning (using the SVM) to classify music based on the vectors and whether or not they represent a match.

### *Support Vector Machine*

The type of machine learner used in this project was a support vector machine (SVM) that compared the values retrieved from MARSYAS. For this project, the SVM known as "SVM<sup>light</sup>" was used. An SVM can be "trained" and used to compare and classify data based on its comprehension of user entered data. To compare data, there is a single classifier in front of each feature vector, 1 or -1, which represents the songs as being either similar or dissimilar, respectively, based on the information gathered from Last.fm and Pandora (see Table 1). The SVM can then learn how comparable values within the vector are based on the classifier. After an SVM is "trained," random vectors from unused songs can then be inputted, and the SVM will output how alike they are based on the comparisons it learned [3].

| Value #                 | Song 1 Value   | Song 2 Value | Difference |
|-------------------------|--|--------------|------------|
| 1                       | .405...  | .399...      | .006...    |
| 2                       | .643...  | .625...      | .018...    |
| 3                       | .218...  | .033...      | .185...    |
| 4                       | .132...  | .123...      | .009...    |
| 34                      | .278...  | .142...      | .136...    |
| <b>Resulting Vector</b> | <b>-1 1:0.006819 2:0.018256</b><br><b>3:0.185454 4:0.009755 ...</b><br><b>34:0.136581 #1-2</b> |              |            |

Table 1: table of sample values used in the feature vector and how the vector would be outputted

## Method

To obtain the social-based data for the experiment, the site Last.fm was chosen because of its large music database and its ease of access through the Last.fm API. Using the Last.fm API, a list of the top 200 recommended songs for each song in the library was retrieved. The number of similar songs to obtain from Last.fm was chosen as 200 because of the limited music library size and the importance that some of the listed songs were in the library. Each recommended song that was available in the library was added to a list of matches. All songs by the same artist were also added to the match list. More matches were found by retrieving a list of all the similar artists to an artist of a particular song and adding all songs by any of those artists to the list of matches.

After figuring out how to find similar songs, an extensive music library was gathered from two group members as well as 54 gigabytes of diverse music from our mentor (O-B in his collection) to ensure variations in music genres. 43 songs were randomly reserved as sample music to run through the SVM after it would be trained. Each of the remaining songs was then put into MARSYAS, in which bextract created a file of feature vectors for all of the songs.

In order to create a complete file of vectors to put into the SVM, each song's values were compared to every other song's values. Since there were about 1990 usable songs, there were a total of around 1,979,055 song comparisons. To compare them all, the absolute value of the difference of each of the 34 values was calculated (see Table 1). Then, each combination of songs was marked as similar or

dissimilar using searches for similar songs and artists from Last.fm and Pandora.

When the comparison file was completed, it was put into the SVM for training. Because of the relatively small music database size and the unequal number of like and unlike songs, the same number of dissimilar and similar songs was picked from the comparison file, resulting in about 11,000-12,000 total comparisons being put into the SVM.

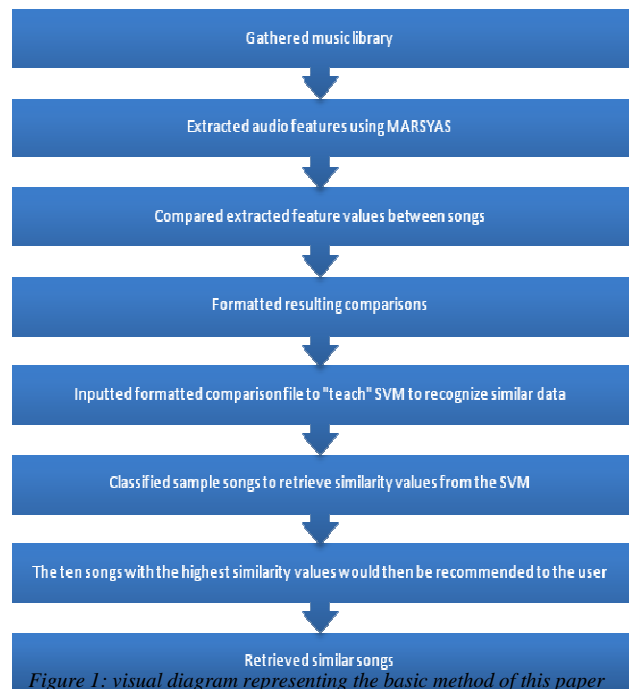


Figure 1: visual diagram representing the basic method of this paper

To evaluate the accuracy of the algorithm, a program was created to conduct an experiment consisting of actual users. Eleven total users were asked to choose one of the sample songs that were set aside earlier for testing (43 sample songs in total). A list of ten random songs from the library was used as a recommendation constant. Then, ten songs similar to the user's chosen song were each retrieved using the paper's discussed algorithm and Last.fm, and six similar songs were taken from Pandora. The user then rated these songs on a scale from 0 to 5, 0 being a highly disliked song and 5 being a highly liked song.

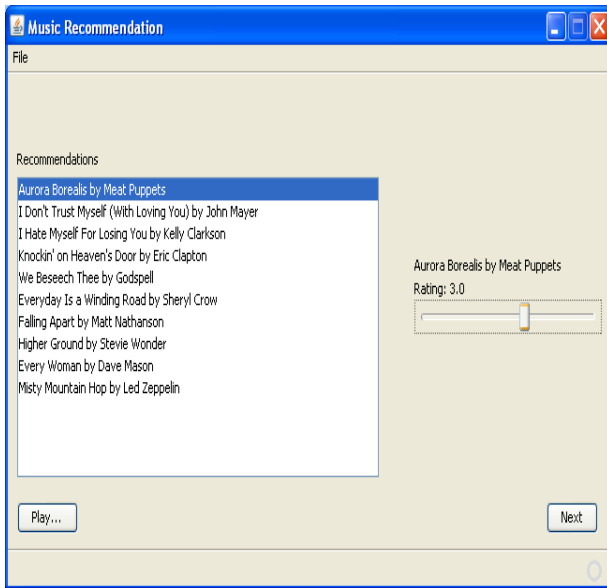


Figure 2: screenshot of the program used for the experiment

The data was analyzed using the Mean Absolute Error (MAE) between the user ratings and the corresponding predicted ratings on a scale of 0 to 5. The MAE is the average of the absolute errors between the user ratings and the SVM ratings. MAE can be represented by the following equation:

$$MAE = \frac{\sum_{i=1}^n |p_i - r_i|}{n}$$

$n$  is the number of recommendations,  $p_i$  is the predicted rating, and  $r_i$  is the user's rating.

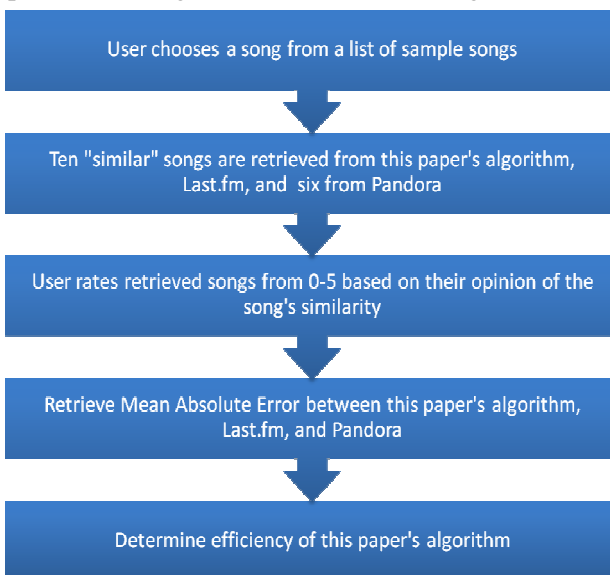


Figure 3: visual diagram representing the experimental process of determining the efficiency of the method

## Analysis

It is generally believed that a hybrid music recommendation algorithm will work better than separate content-based and social-based algorithms because it combines the advantages of both systems. The conducted experiment showed that the method used in this project is comparable to popular music recommendation systems utilized today but lacks sufficient results due to a small music library for testing.

| Music Recommendation System/Site | SVM2  | SVM   | Random | Last.fm | Pandora |
|----------------------------------|-------|-------|--------|---------|---------|
| MAE Value                        | 2.052 | 2.457 | 2.681  | 0.786   | 0.617   |

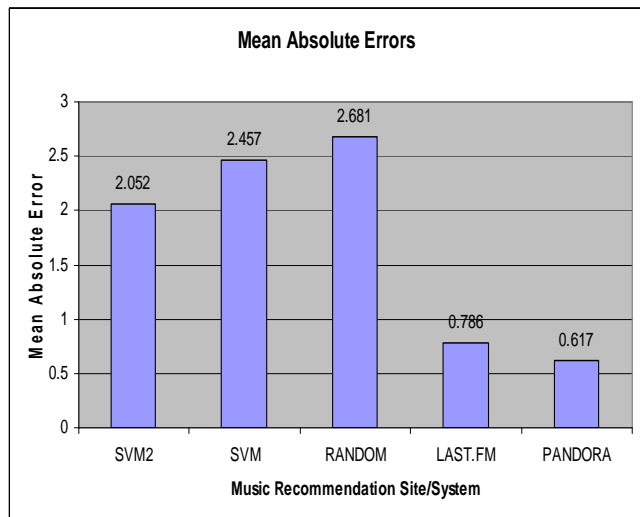


Figure 4: table and graph of experimental results

In the data above, "SVM2" and "SVM" are the mean absolute errors received when comparing user ratings to data retrieved from the SVM. "SVM2" is the final algorithm that this project utilizes, while "SVM" is the old algorithm that is being compared to the newer algorithm purely for statistical data. "SVM2" is error received from the SVM that was trained with the same artists' songs, similar songs from Last.fm and Pandora, and similar artists' songs marked as comparable, while "SVM" is the error received from the SVM that was trained with only similar songs from Last.fm. "Random" is the error obtained from user ratings of ten constant, random songs retrieved from the music library used in this project. "Last.fm" and "Pandora" are the errors collected when comparing user ratings to data

retrieved from both of those music recommendation sites.

According to the results, Last.fm and Pandora had the smallest amounts of error in recommending songs that the user enjoyed while the final algorithm used in this project (SVM2) had the third smallest amount of error. This shows that the design of this project is not too accurate when compared to the current top two music recommendation sites, although the final system was rather accurate when compared to previous algorithms. With a larger music library and more testers, a better result could have been achieved.

## Related Work

As previously discussed, there exist two main bases upon which songs are compared - by use of either social views or content views. The social aspect normally takes into account user ratings of the similarity between songs or like processes. Alternately, the content-based method takes into account certain aspects of a song, such as tempo, pitch, melody, or bass to compare it to other songs.

The ideas for music recommendation systems have been around for many years. In the 1994 article "Social Information Filtering for Music Recommendation," the author introduced the idea that music could be recommended on a social basis. His proposal was that through e-mail, users could rate songs based on their preferences that would then be sent to others for suggestions of music [6]. This was one of the first examples of social-based song recommendation, which was a proposition that would have been difficult to accomplish in 1994 but has already been implemented now. By taking input from Last.fm and Pandora, both music recommendation sites, our music recommendation system does factor in a social-based aspect.

Another recent example of research into use of the content approach was made by a group from the Graduate School of Engineering Science at Osaka University entitled "Content-based Music Filtering System with Editable User Profile." In this group's paper, a music recommendation system was proposed in which one would create a user profile and, by using certain information in the profile, the system would recommend music that would hopefully match with that profile [2]. Rather than matching music to a person in attempts at creating a

music recommender, songs are matched to similar tracks.

A third correlative paper on music recommendation is "A Music Recommendation System Based on Music Data Grouping and User Interests." The paper presented an idea that is more similar to this paper's algorithm than the aforementioned paper. This group proposed analyzing songs with the MIDI format, which is much less commonly used than the MP3 format because of its inferior sound quality and incapability to represent live music. The ventures in this research project are similar to those of this paper because both employ the idea of extracting individual features from songs. The differences occur in this paper's choice to use MP3 formatted songs rather than MIDI formatted songs.

One last related work worth noting for its similarities to this project is one from a group at Phillips Research Europe [1]. In their research work titled "Ensemble Learning for Hybrid Music Recommendation," this group focused on ensemble learning, a process by which multiple artificial learners attempt to study algorithms in order to classify user input. By utilizing the SVM, this project's method implemented the use of machine learning, during which the artificial intelligence program was "taught" to recognize how songs were related based on values extracted from MARSYAS.

## Conclusion

By combining social and content-based music recommendation algorithms, a music recommendation system was created that is somewhat accurate in its user suggestions, as can be seen by the mean absolute error (2.052). When compared to Last.fm and Pandora's errors of 0.786 and 0.617, respectively, the error of the designed system showed that it is not as capable of outputting comparisons similar to the top two music recommendation sites in existence today. However, due to the extremely small library used for this project, the results obtained were not ideal. Had there been more songs, perhaps maybe one million tracks instead of about two thousand, the designed system would have definitely been just as accurate as Last.fm and Pandora.

Despite the large mean absolute error, this system is still advantageous because instead of

combining the outputs of social and content-based music recommendation, it creates the content-based output by utilizing the social-based output. This method also has its disadvantages because there may be issues with combining content and social-based recommendations. If the data from social-based sites are not accurate, that will lead to an inaccurate content-based result, which will ultimately be responsible for ineffective recommendations. However, the main problem with this system is the issue of time inefficiency. If a new song were to be added to the library, that song would need to go through MARSYAS, get its audio feature values extracted and be compared to every other song in the large song library in order to find its similar songs, and then the SVM would need to be retrained, which takes an extensive amount of time. Basically, our method sacrifices speed for accuracy.

As more entertainment moves to the internet, especially in terms of music, our system could be extremely applicable in upcoming or current music sites in order for them to successfully recommend songs to a user. People are increasingly relying on the internet to find things they enjoy, so as peoples' expectations rise, technology must also rise with them. However, music recommendation is far from being perfect. Future work in this area could consist of optimizing both speed and accuracy of hybrid music recommendation, utilizing improved social algorithms (instead of simply pulling similar songs from an existing social music recommendation site's API), and finding different, more effective ways to compare and classify similar songs.

In retrospect, music recommendation is far from perfect, but with the current expanding technology, it will quickly get closer and closer to perfection. This project grabbed areas of existing music recommendation algorithms to create an efficient system, but more research will be required in order to improve music recommendation systems to can appeal to everybody's tastes.

## Acknowledgements

We would like to thank the following people and groups who guided us and gave us the opportunity to work on this research project:

\* Pandora.com and Last.fm

\* Hayden Metsky, for giving up his time to be our knowledgeable mentor.

\* David Vetrano, for his help as our RTA.

\* The NJ Governor's School of Engineering and Technology (Donald M. Brown, Director, Blase Ur, Program Coordinator, & Kristin Frank, Head RTA).

\* The Rutgers University School of Engineering (Dr. Yogesh Jaluria, Outgoing Interim Dean, and Dr. Thomas Farris, Dean).

\* The NJ Governor's School Board of Overseers.

\* The NJ Governor's School of Engineering and Technology 2009's sponsors: Rutgers University, the Rutgers University School of Engineering, the Motorola Foundation, Morgan Stanley, PSEG, Silver Line Building Products, and the families of 2001-2008 program alumni.

## References

1. H. Chen and A. Chen. "A Music Recommendation System Based on Music Data Grouping and User Interests," in the *Department of Computer Science at Tsing Hua University*, 2001.
2. Y. Hijikata, K. Iwahama, K. Takegawa, and S. Nishida. "Content-based Music Filtering System with Editable User Profile," in the *Graduate School of Engineering Science at Osaka University*, 2006.
3. T. Joachims. "SVM<sup>light</sup>: Support Vector Machine," 2008. <http://svmlight.joachims.org/>
4. N. Orio. "Music Retrieval: A Tutorial and Review," in the *Department of Information Engineering at the University of Padova Via Gradenigo*, 2006.
5. G. Percival and G. Tzanetakis. "MARSYAS User Manual for Version 0.2," 2001. <http://marsyas.sness.net>
6. U. Shardanand. "Social Information Filtering for Music Recommendation," in the *Department of Electrical Engineering and Computer Science Archives of MIT*, 1994.
7. M. Tiemann, S. Pauws, and F. Vignoli. "Ensemble Learning for Hybrid Music Recommendation," in *ISMIR*, 2007.