SOUNDSCOUT: A SONG RECOMMENDER BASED ON SOUND SIMILARITY FOR HUGE COMMERCIAL MUSIC ARCHIVES

Peter Hlavac, Brigitte Krenn, Erich Gstrein
Austrian Research Centers GmbH
Studio Smart Agent Technologies
Hasnerstrasse 123, A-1160 Vienna, Austria

ABSTRACT

We present Soundscout, a song recommender operating on huge commercial music archives. In contrast to many playlist generation tools which support the consumer in handling local collections, Soundscout focuses on efficient metadata generation from large bodies of music files. The system automatically generates similarity relations between songs based on acoustic features extracted from the sound files and combines the thus achieved results with a variety of preference models as filters to minimize the risk of inappropriate recommendations.

In the paper we describe how we significantly improved the applicability of state-of-the-art MIR techniques by introducing an efficient k-nearest neighbour algorithm feasible for handling a huge number of tracks. Moreover we introduce filter mechanisms and presentation strategies for improving the acceptability of the sounding similar recommendations. Last but not least we present a Web application of Soundscout for conducting user acceptance tests, and present results from a first exploratory study.

1 INTRODUCTION

Distributing music is currently one of the hottest topics in e- and especially in m-commerce with an enormous market potential. The size of commercial music archives and their short-lived content require efficient and flexible technology. Users of commercial platforms such as Napster [1] or iTunes [2] are confronted with millions of songs (3+ millions at Napster and 3.5 millions at iTunes), and need guidance in order to find songs. Portal operators who offer these services are interested in optimizing the process to increase their sales. In addition, content providers such as SonyBMG, Warner, etc. are interested in extending their Long Tail [3] business by an efficient referring mechanism to non main-stream items.

Common recommender approaches, mainly based on collaborative [5] [6] or item-based filtering algorithms [7] or hybrid combinitions [8] [9], provide some guidance in finding appropriate songs, but have significant shortcommings in supporting the user to explore new territories. While the collaborative approach relies on usage patterns of the community - and therefore tends to represent the main stream - the item-based approach suffers from poor metadata. Audio content providers deliver their audio data together with only some basic information, including artist name, album name, track name, year, genre and pricing information. From a recommender's point of view - where similarity relations between items often form the basis for recommendations - this data quality is insufficient. For instance: Songs are not necessarily similar if they have been created by the same artist and affiliation of tracks to the same genre does not necessarily imply similarity of the songs.

In this situation, music information retrieval (MIR) techniques come in handy, because they help to access information available in the sound file that can be used for similarity computations. The problem for real-life scenarios, however, is that the algorithms are not optimized for huge data sets and that audio similarity between two sound files not necessarily must be perceived or accepted as being similar by the individual user. This may be due to personal preferences and attitudes towards artists, songs, genres, etc. There may be preconceived opinions about artists or songs from different times, etc., for instance: A Robby Williams fan might be disappointed or even offended when provided with a similar-sounding song by a German hit song star. A person who is mainly interested in the latest trends will not appreciate the recommendation of last year's songs. Other people, however, might be inspired. Thus we have not only created a system that scales up to huge data sets but also flexibly provides various filters to sharpen the outcome of similarity computations based on MIR techniques. This way our system becomes more flexible and less dependent on the quality of the individual MIR techniques employed. This approach enables portal operators to improve their services in a flexible and cost-efficient way.

The paper is divided into two parts: In Section 2 we describe our system and in Section 3 we present a Web application of Soundscout that has been implemented for demonstration and user trials (Section 3.1). The results of a first user test are summarized in Section 3.2.

[1] www.napster.com
[2] www.iTunes.com
[3] Long Tail stands for the large number of non top-selling products.

2 SOUNDSCOUT

Soundscout generates for each song a list of similar songs based on: a) the extraction of feature vectors from the audio file, perceptually motivated audio features from the sound file, b) the application of an efficient interpolation algortihm to calcualte similarity values between the feature vectors, c) filtering the thus achieved results with preference models including song and artist popularity, release date and genre incompatibilities. The system is able to operate on any music collection that provides audio files and related metadata such as artist, song title, album name etc. Soundscout initially was developed on a test sample of 60.000 songs to support and improve a personalized mobile music platform called MDP (music download platform) [4]

The objective of Soundscout is not to do basic research in MIR technology or algorithms for user centered playlist generation, but to improve existing techniques such that they can be successfully applied to real-world music archives, in order to support portal operators and content providers in creating high-quality metadata in a flexible and cost-efficient way. Due to that business oriented focus, topics such as performance, content life cycle support and acceptability of recommendations are key issues for the design, development and implementation of the system.

2.1 The Data

From the current MDP track base of about 2 million songs a subset of the 60.000 most downloaded tracks was taken. The original audio files are encoded in aac format with 64kbit sampled at 32kHz and are converted to standard pcm encoded wav files with a sample rate of 22. 05khz and 16bit of dynamic range. The metadata available are song title, artist and album name, an “explicit tag” indicating explicit lyrics, the release year, and genre information with one song typically belonging to various genres.

2.2 Feature Extraction from Audio Files

For extracting perceptually motivated audio features from the song files, we employ state-of-the-art MIR technology, in particular [4] for extracting timbre information using Gaussian Mixtures of MFCCs. We currently also experimet with an approach described in [2] [5] . Employing the algorthm of Pampalk et al., the average computation time for extracting audio features of a song, including the conversion to a wav file, is about 30 seconds on a Pentium 4 running at 3Ghz with 2GB RAM. The feature extraction process for all 60.000 tracks thus took three weeks on a single machine. When only analyzing 30 seconds previews (manually sliced by the record companies) instead of using the whole audio file, the same task requires 6 days (8 seconds per snippet).

For analysing the 60.000 30 seconds snippets employing Musicminer the task was completed within two days distributed on three machines.

2.3 Finding Similar Tracks

To find the most similar songs to a given one, all songs (feature vectors) have to be compared and then ordered by their distance to the given track. This works fine with small music collections, but is impracticable for huge music libraries, because the complexity of computing all pairwise distances $O(n^2/2)$, where n is the number of tracks. In other words: Calculating the distance between two songs takes about 50 milliseconds using the techniques described in [4]. This makes 20 songs per second. A subset of 60.000 tracks would need $(60.000^2)/2 = 1.800.000.000$ comparisons, which would require on one machine about 2,8 years. To compute all distances between songs from a music archive like iTunes or Napster with more than 3 million tracks, the machine would have to live longer than 7135(!) years to complete the job.

2.4 The Interpolation (IP) Algorithm

To overcome such time investments, the foremeost goal was to drastically reduce the number of similarity computations. Our first approach was to take a subset of songs (=prototypes) and compute their distances to each track in the archive. This reduces the number of comparisons from $O(n2/2)$ to $O(n * \#prototypes)$. Now each song is assigned to its nearest prototype. The music archive thus is split into clusters of songs each represented by one prototype. If a song is chosen by the user, random selected songs from the assigned cluster are recommended. It turned out that the recommendations were too heterogenious and in many cases far from sounding similar to the chosen song. As a consequence we have developed a more sophisticated algorithm inspired by multidimensional scaling [10]. It uses prototypes (=sampling points) to interpolate the distance between two songs. The algorithm computes the 500 nearest songs to each track with a reduced complexity of $O(n)$, where n is the number of tracks.

2.4.1 Get nearest Prototypes

A subset of k (k=400) randomly chosen prototypes is taken from the music archive. For each song the nearest l prototypes ($l < k$, currently $l = 5$) are computed and stored in a database. For 60.000 tracks that classification took about one week on two clustered machines ($= 400 * 60000 = 24 * 10^6$ distance computations). This is still a time consuming task but has linear complexity $O(n)$. The computation time for finding the nearest prototypes for a music archive such as iTunes with 3+ million tracks is about one month on a cluster with 25 machines processing in parallel.

[4] MDP is a product of the mobile services provider 3United Mobile Solutions AG, a VeriSign company. See http://www.3united.com/. Applications of MDP are currently online in Europe, Asia and the USA.

[5] See also http://musicminer.sourceforge.net

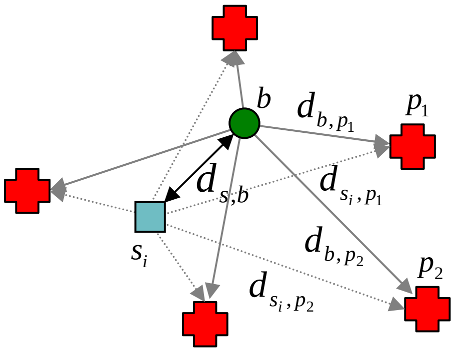

Figure 1. shows an interpolation of the distance between two songs with the help of 5 sampling points (prototypes)

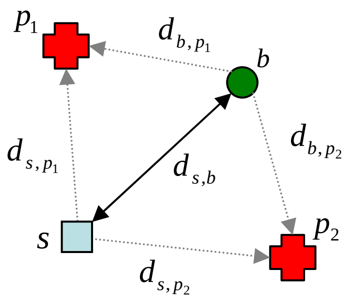

Figure 2. Using the IP algorithm may lead in wrong distance values

2.4.2 *Interpolation using Prototypes*

To find similar tracks to a given base item b the computed l nearest prototypes p_j are selected from the database. A query is executed to get all songs s_i that have those l prototypes as nearest neighbours. The interpolated distance d_{b,s_i} from the base song to the song s_i is illustrated in Figure 1. It is computed with the following equation, where the value of d_{b,p_j} is the distance from the base song b to the prototype p_j and d_{s_i,p_j} is the distance from the song s_i to the prototype p_j.

$$d_{b,s_i} = \sqrt{\sum_{i=1}^{5}(d_{s_i,p_j} - d_{b,p_j})^2} \quad (1)$$

After interpolating the distances from all matching songs to the base song, the nearest n songs (currently up to 500, if available) are stored in the database. While the main effort of the algorithm lies in finding the intersection of common prototypes, the size n has only marginal impact on the performance. It took one week on a single machine to compute the nearest neighbours to each of the 60.000 songs. However, the IP algorithm could lead to wrong distance values, as illustrated by the pathological example in Figure 2. Even though the actual distance between b and s is greater than 0, the Euclidian distance formula with equal distances from the prototypes to the songs results in a zero.

$$d_{b,s_i} = \sqrt{(1-1)^2 - (1-1)^2} = 0 \quad (2)$$

Such a case is more likely to occur when only two prototypes are used. In general, the more common prototypes are used the more adequate are the computed distances, but the fewer songs exist that share those prototypes. When developing Soundscout it has turned out that 5 protoypes are a suitable number for a set of 60.000 tracks.

2.5 Improving Recommendation Quality through Preference Models

Finding songs with similar acoustic features does not necessarily imply good recommendations. There are many reasons why people do not accept particular two songs as being similar. For instance: It may be rather frustrating for a user, when Soundscout suggests songs that are not known, not popular or incompatible with certain audiences' life style, selfconception etc. To minimize the risk of inappropriate recommendations additional filter mechanisms are called for. For the current version of Soundscout we have implemented a popularity, a time and a genre exclusion filter to trimm the list of the computed nearest 500 songs. Employing such a defensive approach may leave us with only a small number of remaining songs out of the 500, which is still better than providing unacceptable recommendations. As regards the filters, it is important to note that the suitability of individual filters depends on the content of the song library, the audience addressed, and the way recommendations are presented to the user, cf. the description of "recommendation groups" in section 3.1. The longterm goal is to equip Soundscout with a library of filters that can be flexibly added to and discarded from the running system, and thus support portal operators in optimizing their services.

In the following we describe three filters that have shown to be useful on the MDP data set and the three recommendation groups: songs from the "Same Artist", the "Same Genre" and "Mixed Genres".

2.5.1 *Popularity Filter*

Our analyses of user behaviour on the music portals powerd by MDP showed that 80% of all track relevant actions (i.e. visiting the page where the song is presented) were performed on only 2% of all available songs. This fact inspired us to implement a 'popularity filter' that takes the Web presence of artists and songs into account. In particular, we take the Google page count g_i as "mainstream" value for a given song i; g_i is retrieved by querying the search engine with the expression <lyrics "ARTIST_NAME SONG_NAME">, e.g. <lyrics "Nelly Furtado Say it right">. Based on the song popularity values an artist popularity a_p is computed. This is the sum of all song popularity values for that artist divided by the number of songs from the artist.

$$a_p = \frac{\sum_{i=1}^{n} g_i(a)}{n} \quad (3)$$

Songs that are below a given artist or song popularity threshold are not recommended by Soundscout. The threshold differs in the three recommendation groups starting with low values of song popularity in the group of songs from the same artist, employing increasing values of both artist and song popularity in the group of songs of similar genres up to songs of mixed genres. This restrictive handling of the 'populartity filter' was used for Soundscout's evaluation application, but will be extended to a user defined filter for fine tuning of the recommendation results. (e.g. "Give me unknown similar sounding song to Hung U of Madonna")

2.5.2 Time Filter

Another approach to improve the default recommendation quality of Soundscout is taking into account the creation date of tracks, assuming that this reflects, to some extent, the zeitgeist represented by a piece of music. This filter has in the first place been designed for recommendations from Mixed Genres to avoid bothering a user looking for similar songs from the latest charts with similar sounding oldies. Or to put it the other way round: to spare a user looking for songs similar to Frank Sinatra songs a recommendation of the latest production from R. Kelly because of sound similarity.

2.5.3 Genre Exclution Filter

A genre exclusion filter is implemented to avoid that Soundscout would for example recommend techno songs to a chosen R&B song. This filter, in its simplest version, strongly depends on the metadata provided by the content provider which often is dreadful as most content typically is tagged with pop or rock. In such a constellation the system relies on the results from the acoustic analysis, and knows that the results are better presented under the heading "Broaden my Horizon" than under "Similar Songs".

3 EVALUATION

3.1 The Web Interface

In the following we describe the web interface implemented for user trials. [6] When visiting the page, users can search for songs by artist name, song name or by chosing from predefined genres. On the results page, a list of songs matching the search criteria is displayed (see Figure 3). The user can play a 30 seconds preview of the songs by clicking the play icon next to the song or get recommendations of similar sounding songs by clicking the notes icon. Similar sounding songs are presented in three recommendation groups: songs from the Same Artist, songs of the Same Genre, and songs of Mixed Genres. Visitors may browse through Soundscout instantly and give feedback at any time they like or they can register at the start page. In the latter case they are forced to vote every time when they listen to a recommended song. The voting options are "sounds similar", "quite similar", "not really" and "noo!", see sound.vote on the right of the screenshot in Figure 3. For evaluation, we group the votings into positive (sounds similar, quite similar) and negative (not really, nooo!) ones.

[6] The Web demonstrator of Soundscout is accessible on http://soundscout.researchstudio.at.

3.2 Evaluation Results

For evaluation, we looked at a period of one month (December 2006 to January 2007). During that time the webpage was visited 223 times. The visitors listened to 971 snippets and voted 484 times. 53 people registered, with the majority from Austria (37), followed by Great Britain (11), Slovenia (1), Pakistan (1) und Brazil (1). Those users were listening to 202 songs and they voted 182 times with 4 votes on average per user.

3.2.1 All Users

Vote Options	Votes	Pos/Neg Votes	
sounds similar	144		
quite similar	170		
		314	**64.88%**
not really	107		
noo!	63		
		170	**35.12%**
Total Votes	**484**	**484**	

Table 1. Distribution of votes by all users

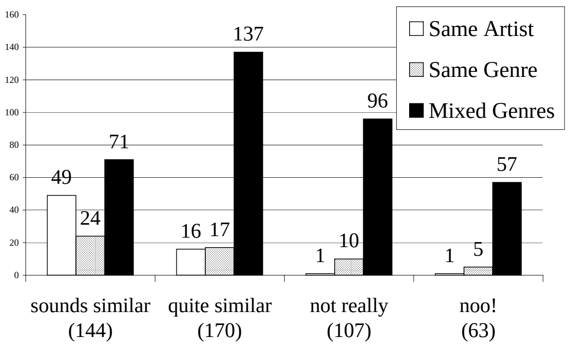

Figure 4. Distribution of all votes according to "Same Artist", "Same Genre", "Mixed Genres")

The majority of users were pleased with the recommendations as can be seen in Table 1. Approximately two third of the votes were for "sounds similar" or "quite similar" and only one third of the votes was for "not really"

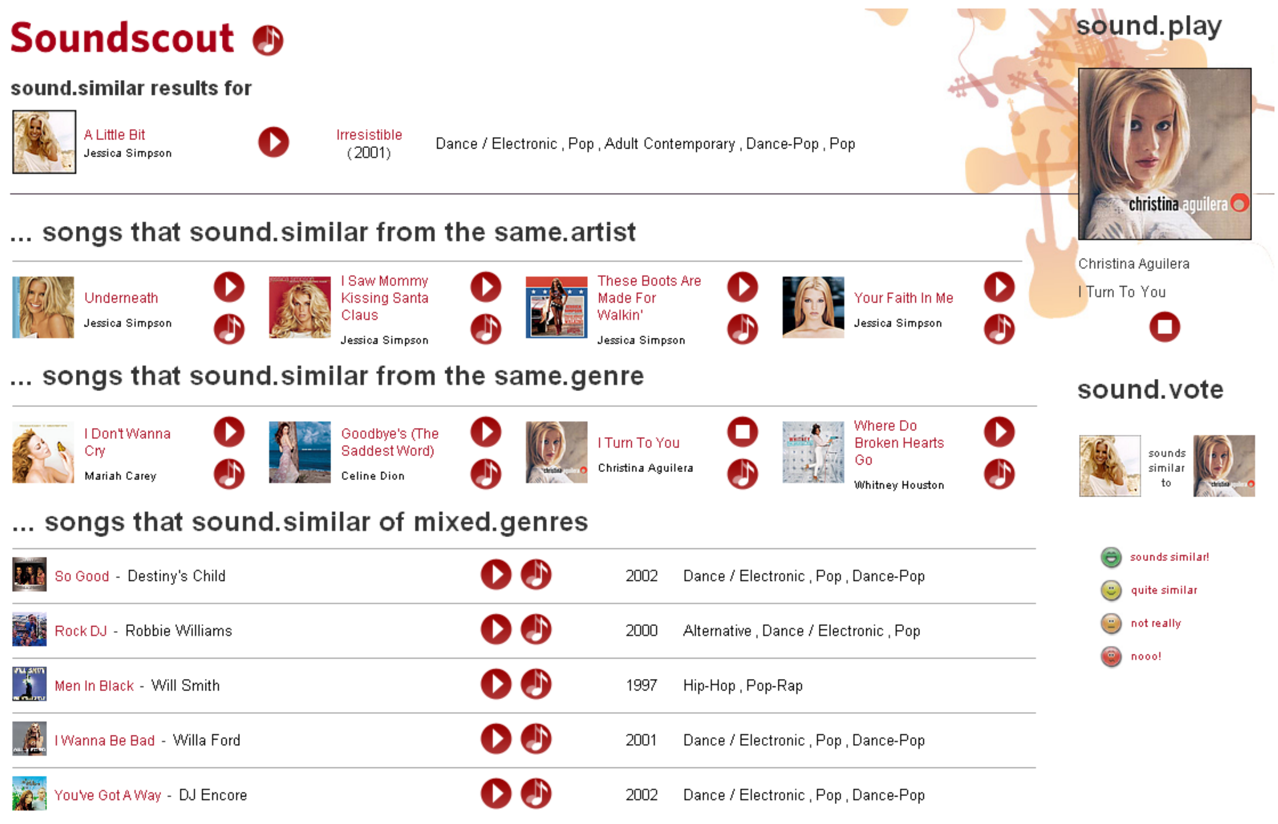

Figure 3. Soundscout Interface: Recommendations in the left frame, Snippetplayer (sound.play) and Voting Box (sound.vote) in the right frame

or "nooo!". Figure 4 shows the votes per recommendation group. The group "Mixed Genres" has the most votes (361 totals). We take this as an indicator that users are most interested in recommendations that are from this group, as the overall user activity (recorded clicks) in this group is as twice as high as in the other groups. The distribution of votes (208 positive versus 153 negative), however, shows that recommendations from mixed generes are the hardest ones, whereas songs from the same artist are much more likely to be considered as sounding similar with only two negative votes compared to 65 positive ones. The ratio positive versus negative votes diminishes from the group "Same Artist" (97% positive votes) to "Same Genre" (73% positive votes) to "Mixed Genre" (58% positive votes).

3.2.2 *Registered Users*

To become a registered user, visitors have to fill in their birth date, gender, affiliation to music, and the country they are come from. All in all, registered users were more critical than unregistered ones. Only 45% of the votes were positive, see Table 2. As the distribution of the votes over the recoomendation categories (Figure 5) shows that the users again were pleased with recommendations from the groups "Same Artist" and "Same Genre" with 96% positive votes for the former and 67% positive votes for the latter. Once more recommendations from "Mixed Genres" turned out to be problematic with only 47% positive votes. However, there is a gender difference: females voted less strict (77% positive versus 23% negative) than males (53% positive versus 47% negative). See Table 3.

Vote Options	Votes	Pos/Neg Votes	
sounds similar	38		
quite similar	61		
		99	**54.40%**
not really	51		
noo!	32		
		83	**45.60%**
Total Votes	**182**	**182**	

Table 2. Distribution of votes by registered users

	Votes	**Pos**	Votes	**Neg**	Total
41 men	89	**52.66%**	80	**47.34%**	169
11 women	10	**76.92%**	3	**23.08%**	13
					182

Table 3. Distribution of votes by gender

4 CONCLUSION

We have presented Soundscout an application that generates high quality metadata based on sound similarity to support portal operators and content providers to im-

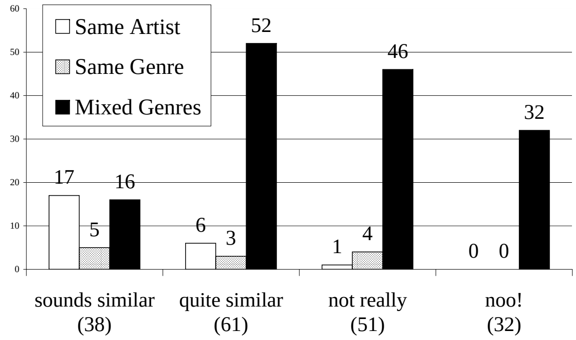

Figure 5. Votes by registered users acording to ”Same Artist”, ”Same Genre” and ”Mixed Genres”

prove their services. Due to its architecture supporting parallel computation and an ennfficient strategy - the *Interpolation Algorithm* - for solving the k nearest neighbour problem Soundscout is feasable for handling huge music archives of the commercial world. We significantly improved the quality of sounding similar recommendations by introducing filters such as a popularity, a genre exclusion and a time filter minimizing the risk of controversial recommendations. Promising results were drawn from an early evaluation which showed that a majority of users is satisfied with the recommendations from Soundscout. From 484 votes total, the voting options “sounding similar” and “quite similar” were clicked 314 times which is a ratio of **65% positive** to **35% negative votes**.

5 FURTHER DEVELOPMENT

First experiments with alternative audio features and MIR technologies [2] showed promizing results especially in clearing weaknesses of the quality of sound simiularity of our currently applied MIR technology. Furthermore, a wider range of filters, especially including psychologically motivated preference models such as STOMP [11], will be implemented and offered to the user to fine-tune the results. While writing the paper a suitable process has been devised to integrate Soundscout with the currently deployed version of MDP.

6 REFERENCES

[1] Knees, P., Pohle, T., Schedl, M., and Widmer, G. ”Combining audio-based similarity with web-based data to accelerate automatic music playlist generation”, *In Proceedings of the 8th ACM international workshop on Multimedia information retrieval.*, New York, USA, 2006.

[2] Mörchen, F., Ultsch, A., Nöcker, M., and Stamm C. ” Databionic visualization of music collections according to perceptual distance”, *Proceedings 6th International Conference on Music Information Retrieval (ISMIR 2005).*, London, UK, 2005.

[3] Mörchen, F., Ultsch, A., Nöcker, M., and Stamm C. ”Visual mining in music collections”, *Proceedings of the 29th Annual Conference of the German Classification. Society (GfKl)*, Magdeburg, Germany, 2005.

[4] Pampalk, E., Dixon, S., and Widmer, G. ”On the Evaluation of Perceptual Similarity Measures for Music”, *In Proceedings of the 6th International Conference on Digital Audio Effects.*, London, UK, 2003.

[5] Herlocker, J., L., and Konstan, J., A., Terveen, L., G., and Riedl, J., T. ”Evaluating Collaborative Filtering Recommender Systems”, *ACM Transactions on Information systems, Vol. 22 No. 1, January 2004, Pages 5 53* , New York, USA, 2004.

[6] J. Herlocker, J.A.Konstan, J. Riedl ”Explaining Collaborative Filtering Recommendations”, *ACM 2000; Conference on Computer-Supported Collaborative Work.*, New York, USA, 2000.

[7] Meteren, R. v., Someren, M. v. ”Using Content-Based Filtering for Recommendation”, *In Proceedings of ECML 2000 Workshop: Machine Learning in New Information Age.*, Barcelona, Spain, 2000.

[8] Gstrein, E. et al. ”Adaptive personalization: A multi-dimensional approach to boosting a large scale mobile music portal.”, *In Fifth Open Workshop on MUSICNETWORK: Integration of Music in Multimedia Applications.*, Vienna, Austria, 2005.

[9] Burke, R. ”Hybrid recommender systems: Survey and experiments”, *In User Modeling and User-Adapted Interaction.*, 2002.

[10] Kruskal, J. B., and Wish, M. ”Multidimensional Scaling”, *Sage University Paper series on Quantitative Application in the Social Sciences.*, Beverly Hills, USA, 1978.

[11] P.J. Rentfrow, S. Gosling. ”The Do Re Mi’s of Everyday Life: The Structure and Personality Correlates of Music Preferences”, *Journal of Personality and Social Psychology. Vol. 84, No. 6, 1236-1256*, 2003.