MUSICSUN: A NEW APPROACH TO ARTIST RECOMMENDATION

Elias Pampalk and Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST)
IT, AIST, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

ABSTRACT

MusicSun is a graphical user interface to discover artists. Artists are recommended based on one or more artists selected by the user. The recommendations are computed by combining 3 different aspects of similarity. The users can change the impact of each of these aspects. In addition words are displayed which describe the artists selected by the user. The user can select one of these words to focus the search on a specific direction.

In this paper we present the techniques used to compute the recommendations and the graphical user interface. Furthermore, we present the results of an evaluation with 33 users. We asked them, for example, to judge the usefulness of the different interface components and the quality of the recommendations.

1 INTRODUCTION

Popular music recommendation services include Amazon's personalized recommendation lists, personalized Internet radio (e.g. Last.fm and Pandora), and services that make music blogs more accessible (e.g. the Hype Machine). [1] In contrast to most existing recommendation services MusicSun uses neither collaborative filtering nor manually annotated data. Instead we use techniques to automatically extract information from audio and web pages.

MusicSun is basically a query-by-example interface. A user selects one or more artists and is given a list of similar artists. In addition, MusicSun offers the users several options to modify the recommendations. In particular, the users can (1) choose which vocabulary the interface uses to describe artists, (2) select a word from a list of words summarizing the query which they consider most relevant to their search, and (3) choose which aspects of similarity they are most interested in (the options are: audio-based sound similarity, web-based sociocultural similarity, or similarity with respect to the selected word).

Related work in terms of discovering music includes a number of very different approaches. One example is the query-by-example interface (for songs) presented in [1] where an "aha" slider can be used to filter results from the same genre and obtain more interesting recommendations. Related work also includes work on visualizing music collections, enabling the user to easily browse and discover new music, e.g., [5, 7, 8, 9, 10, 12, 15, 16]. In terms of discovering music a notable approach is the work presented in [3] which efficiently combines web services to give high quality music recommendations. MusicSun is closely related to the MusicRainbow user interface [14] with respect to the techniques used and the focus on discovering new artists. However, MusicSun gives the users more options to focus their search and also requires them to make more choices.

[1] http://last.fm, http://pandora.com, http://hypem.com

2 TECHNIQUES

The MusicSun interface is generated using audio (tracks) and the associated artist names as input. Using the artist names we crawl the web (using Google), and parse the retrieved pages using 4 different vocabularies. Using this data extracted from the web we compute two of the three aspects of similarity we use and summarize individual artists and groups of artists with words. Using the tracks and audio similarity techniques we compute the third similarity aspect. In this section we briefly describe these techniques and how we combine them. Finally, we list some example recommendations.

2.1 Web-Crawling, Vocabularies, and Summarization

To extract "community information" from the web we use the approach suggested in [18]. For each artist we query Google using the artist's name and the terms "music" and "review" as constraints. We retrieve the top 50 ranked pages per artist and parse them for any words they contain.

We parse the retrieved pages using 4 manually compiled vocabularies containing words suitable to describe music (see Table 1). The vocabularies are based on our previous work [14] where we used 3 vocabularies which implied a hierarchy of concepts.

For each artist we count how often words from the vocabularies occur on the retrieved web pages and compute the $tfidf$ weight for each word (see [13] for a more detailed description of the specific implementation we use). Using these weights we compute similarities as described in the next section. In addition we use this information to select words to summarize each artist. We apply the summarization technique suggested in [6]. The basic idea is to select words that not only occur frequently on the respective web pages, but also are suitable to distinguish the given set of artists from all other artists (for further details see [13]).

There are mainly two limitations to this approach. First of all, we assume that artist names are unique identifiers. However, many artist have ambiguous names. Furthermore, the approach only works for artists which are mentioned on a larger number of web pages.

Vocabulary	Examples	Words
Genre/styles	rock, alternative rap, sunshine pop	255
Instruments/types	jazz guitar, female, orchestra	167
Moods/adjectives	smooth, angry, contemporary	452
Countries/regions	Afro-Cuban, Nashville, European	93
	total number of words:	967

Table 1. Description of the 4 vocabularies used.

2.2 Similarity

MusicSun's recommendations are based on combining 3 similarity aspects. By default these are weighted equally. However, their weight can be adjusted using sliders. The first aspect is sound similarity which we compute by analyzing audio contents. The second aspect is sociocultural similarity which is computed by analyzing web pages. (The sociocultural similarity is high if the artist names occur on the same web pages, or if the same words are used on pages they occur on.) Third, we compute the similarity with respect to the word the user has selected to focus the search in a specific direction.

Audio-based similarity: The artist similarity is computed as suggested in [14] which is based on computing the similarity of tracks as described in [11]. To compute the distance between a set of artists (the user's query) and artist X (a possible candidate for a recommendation), we compute the average distance of X to all artists in the set.

Web-based similarity: As described above we compute the $tfidf$ weighting. The distance between length normalized $tfidf$ vectors is computed using the Euclidean norm. To compute the distance between a set of artists and artist X, we compute the average distance of X to all artists in the set.

Word-based similarity ("sun ray"): One of the main features of the user interface is the option to focus the search by selecting one of 9 words that summarize the query artists. For example, when searching for artists similar to ABBA the user can choose to focus the search, for example, on "pop" or "Swedish". (One of the 9 options is always randomly chosen by default.) The similarity with respect to the selected word is computed based on the $tfidf$ weighting. For a given word, the most relevant artist is the one with the highest $tfidf$ weighting. (Thus the word-based similarity is independent of the query artists.)

Combined similarity: First, given the query we compute the similarity of every recommendation candidate according to each of the three similarity aspects. Second, we aggregate the similarity ranks by computing the weighted average of the rank of each artist. Alternatively, this could be done by first normalizing the computed similarities and by combining them linearly (see e.g. [2, 12]). The main advantage of using the ranks is that no normalization is needed.

2.3 Recommendation Examples

Table 2 shows some example recommendations. For each of the 4 examples one of the 9 words summarizing the artist is selected (and shown to the right of the artist's name). The similarity weights were set to their default values (all weights equal). The top 5 recommendations are shown. For example, Jorge Ben is the fifth recommendation when using Gilberto Gil as query and setting the focus on "political". [2] Changing the focus (e.g. in the case of Madonna from "pop" to "singing") does not completely change the results because the audio and web-based part of the similarity computation (2/3 of the weight in the default settings) still produce the same results.

Madonna → Pop	Madonna → Singing
1 Britney Spears	1 Whitney Houston
2 Lisa Stansfield	2 Lisa Stansfield
3 George Michael	3 Maria Carey
4 Whitney Houston	4 Britney Spears
5 Maria Carey	5 Macy Gray

Eminem → Controversial	Gilberto Gil → Political
1 Death Row	1 Caetano Veloso
2 Black Eyed Peas	2 Chico Buarque
3 The Streets	3 Caetano Veloso & Gilberto Gil
4 Dr. Dre	4 David Byrne
5 Ice-T	5 Jorge Ben

Table 2. MusicSun example recommendations.

3 USER INTERFACE

The components of the MusicSun interface are shown in Figure 1. A demonstration video is available online. [3] The user starts by entering an artist name in the search box. When the user activates the search box a list slides in from the left. After finding an artist the user drags it into the sun (a circle in the center of the screen), the search box slides away, and the main elements of the interface slide back into position and are displayed as shown in the Figure 1.

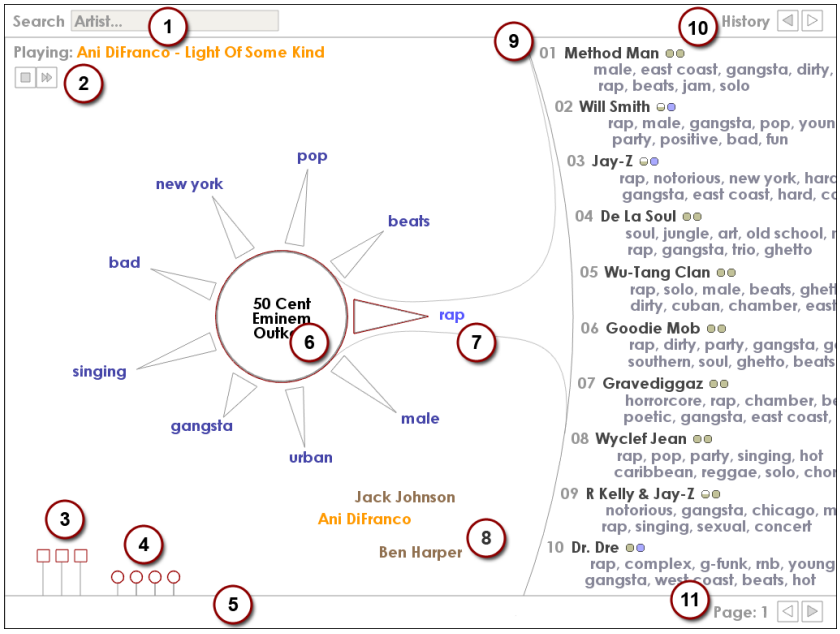
As soon as one or more artists are located in the sun the *rays* are labeled with words to summarize the query artists. The triangular shape of each ray encodes the following information with respect to the word it represents: If the side of the triangle facing the sun is longer, then the respective word describes the artists better. If the length of the ray is longer, then there are more artists in the collection which can also be described using the respective word. One of the rays is randomly preselected by the system. The user can select a different ray by clicking on it. Once a ray is selected, it spins itself into the rightmost position, indicating that it is currently being used to modify the recommendations.

The user can listen to artists by clicking on their names. A second click plays the next song from the same artist. From each song only a 20 second excerpt is played which is selected by the RefraiD chorus detection function [4]. The currently playing song is displayed in the upper left area together with simple playback controls.

The user can drag artists from the recommendation list into the sun, or drag them to the area surrounding the sun

[2] Jorge Ben is often described as less political than all his contemporaries. This is an example where a negation loses its meaning when focusing only on individual words.

[3] http://pampalk.at/musicsun/



1 - Search box to enter (partial) artist names via the keyboard. A search window slides in from the left when activated.
2 - Basic audio playback controls (stop & skip song). Playback starts by clicking on an artist.
3 - Sliders controlling the weights on each similarity aspect.
4 - On/off switches for the vocabularies. In the screenshot all switches are on.
5 - Mouse over help text is display here. This information is particularly important for (3) and (4).
6 - Query artists are placed inside the sun.
7 - The user can choose one of 9 rays. In the screenshot "rap" is selected.
8 - Storage area: the users can place artists here which they want to remember or keep out of the recommendation list.
9 - List of recommendations
10 - History buttons (undo/redo actions)
11 - Previous/next page of recommendations

Figure 1. Screenshot of the MusicSun user interface with 50 Cent, Outkast, and Eminem selected as query.

("storage area"). This storage area can be used, for example, to keep track of previously found artists. Artists can also be dragged back into the recommendation list.

The two small circles displayed next to each artist in the recommendation list encode information about the number of pieces used for the audio analysis (first circle), and the number of web pages found (second circle). An empty circle indicates that the system did not have enough data, thus the recommendation is probably flawed. A half empty circle means that the results are probably questionable. A full circle means that the amount of data used is about average. In case a Google search for an artists yielded an extremely high number of pages the circle is shown in a blueish color. This might indicate either a very famous artist, or an ambiguous name.

In the recommendation list each artist is described with some words. These words are computed in the same way as the rays. More relevant words are located on the left. If the user changes the vocabularies, then the rays as well as these artist summaries are recomputed to contain only words belonging to respective vocabularies. If the user changes the slider settings the recommendation list is instantly recomputed.

4 EVALUATION AND RESULTS

To evaluate the MusicSun interface we asked 33 volunteers to try out the interface. 16 males and 17 females participated with an average age of 28.7 years (standard deviation = 8.4). All participants used computers on a daily basis. 27 participants had previously never used a tabled PC. 15 had musical training. 24 stated that they did not spend any time in the last month actively searching for new music. 10 primarily discovered artists through recommendations from friends. The second most frequently recommendation source mentioned was radio followed by movie soundtracks. All participants had a musical taste that partially overlapped with the contents of the music collection we used in the evaluation.

	Hard	=	Easy	NA
Learning to use MusicSun	4	1	28	0
Discovering artists	2	0	26	5
	Low	**=**	**High**	**NA**
Quality of the recommendations	3	2	27	1
Quality of the artist summaries	4	3	16	10
Fun factor	1	1	30	1
Interest in future usage	3	1	29	0
Usefulness of optional components:				
small circles in recommendation list	13	4	11	5
similarity sliders	9	4	19	1
information encoded in ray shapes	8	2	20	3
choosing vocabularies	6	1	24	2

Table 3. Evaluation results: For all questions we used a 7-point scale. Users who answered by choosing one of the 3 points on the right or the left of the scale are grouped into one category. The number of users who selected the exact center of the scale is marked with an equal sign. In addition, users had the option not to answer a question which is marked with "NA".

The participants were asked to use a tablet PC and pen as input device. The implemented interface was created based on a collection of 999 artists from various genres. While trying out the interface we asked the users questions which are summarized in Table 3. In average the participants spent about 20 minutes with the interface. In the remainder of this section we summarize some of the qualitative findings.

Describing the unknown: For each artist in the recommendation list a few words summarizing the artist are displayed right under the artist's name. We asked the participants to rate the quality of these summaries. 16 users

said the quality was good. However most users ignored these summaries when using the interface. 10 users were not able to answer the question because they never looked at them. Instead of reading the summaries they searched for artist names they were familiar with. In the vicinity of known artists they searched for unknown artists. In addition, an important factor in choosing which of the unknown artists to listen to was the interestingness of the respective name.

Usefulness of reliability indicators: The small circles next to each artist in the recommendation list indicate how reliable the recommendations are. Overall, the users considered this information least useful. Most users did not notice them until we pointed them out in the interviews. However, a few users mentioned that they found it useful to know if a specific artist is very famous (which correlates with the number of web-pages found), and how many pieces per artists were in the database. Only in one case a user noticed that the circles actually served their purpose (to explain a failure in the recommendations). In this particular case the artist's name was "Chess" and the circle indicating the number of web pages was blue.

Similarity weight sliders: Most users tried out the sliders briefly and then ignored them. One user found the results in the recommendation lists better when the focus was on web-based similarity. Another user said the same for audio-based similarity. However, generally it was not obvious to the users what the differences in the rankings were when focusing on either one of the two.

The word based slider was extensively used by three users to focus their search on, for example, Italian music (this was done by increasing the weight of the word similarity slider and selecting the respective ray). However, reliably extracting country information from unstructured web pages is difficult. For example, a frequent occurrence of the word Italy could also mean that the artist held a concert in Italy.

Vocabulary chooser: In contrast to the similarity sliders the users found the consequences of choosing a different vocabulary set easy to understand. Some users focused only on country and region names, others only on genres and styles.

Requested features: A few users asked to be able to select more than one ray at once and to be able to select rays permanently (one of the biggest problems with the interface is that the rays are recomputed when the artists inside the sun change, sometimes this means that a previously selected word disappears).

5 CONCLUSIONS

We presented a new approach to artist recommendation which combines information extracted from web pages with information extracted from audio. We built a new query-by-example(s) user interface which allows the user to control several recommendation parameters. We conducted an evaluation with 33 users. From the feedback we conclude that there are two main directions for further improvements of the user interface: finding better ways to link unknown artists with known artists, and enabling the user to select more than one word to focus the search on.

Acknowledgments:
This work was supported by CrestMuse, Crest, JST.

REFERENCES

[1] J.-J. Aucouturier & F. Pachet, "Music similarity measures: What's the use?" in *ISMIR*, 2002.

[2] S. Baumann, T. Pohle, & V. Shankar, "Towards a socio-cultural compatibility of MIR systems," in *ISMIR*, 2004.

[3] O. Celma, M. Ramírez, & P. Herrera, "Foafing the music: A music recommendation system based on RSS feeds and user preferences," in *ISMIR*, 2005.

[4] M. Goto, "A chorus-section detection method for musical audio signals and its application to a music listening station", in *IEEE Transactions on Audio, Speech, and Language Processing*, Vol.14, No.5, pp.1783-1794, 2006.

[5] P. Knees, M. Schedl, T. Pohle, & G. Widmer, "An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web," in *Proc of ACM Intl Conf on Multimedia*, 2006.

[6] K. Lagus & S. Kaski, "Keyword selection method for characterizing text document maps," in *Proc of Intl Conf on Artificial Neural Networks*, 1999.

[7] P. Lamere, "Search inside the music," Sun Microsystems Laboratories, Tech. Rep., 2006, http://blogs.sun.com/plamere/resource/sitm_two_pager.pdf.

[8] F. Mörchen, A. Ultsch, M. Nöcker, & C. Stamm, "Databionic visualization of music collections according to perceptual distance," in *ISMIR*, 2005.

[9] R. Neumayer, M. Dittenbach, & A. Rauber, "PlaySOM and PocketSOMPlayer, alternative interfaces to large music collections," in *ISMIR*, 2005.

[10] E. Pampalk, "Islands of Music: Analysis, organization, and visualization of music archives," MSc thesis, Vienna University of Technology, 2001.

[11] E. Pampalk, "Computational models of music similarity and their application in music information retrieval," PhD thesis, Vienna University of Technology, 2006.

[12] E. Pampalk, S. Dixon, & G. Widmer, "Exploring music collections by browsing different views," in *ISMIR*, 2003.

[13] E. Pampalk, A. Flexer, & G. Widmer, "Hierarchical organization and description of music collections at the artist level," in *Proc of European Conference on Research and Advanced Technology for Digital Libraries*, 2005.

[14] E. Pampalk & M. Goto, "Musicrainbow: A new user interface to discover artists using audio-based similarity and web-based labeling," in *ISMIR*, 2006.

[15] I. Stavness, J. Gluck, L. Vilhan, & S. Fels, "The MUSICtable: A map-based ubiquitous system for social interaction with a digital music collection," in *Intl Conf on Entertainment Computing*, 2005.

[16] M. Torrens, P. Hertzog, & J.-L. Arcos, "Visualizing and exploring personal music libraries," in *ISMIR*, 2004.

[17] R. van Gulik, F. Vignoli, & H. van de Wetering, "Mapping music in the palm of your hand, explore and discover your collection," in *ISMIR*, 2004.

[18] B. Whitman & S. Lawrence, "Inferring descriptions and similarity for music from community metadata," in *Proc of Intl Computer Music Conf*, 2002.