

A Tool to Support Finding Favorite Music by Visualizing Listeners' Preferences

Satoko Shiroy
Department of Computer Science
University of Tsukuba
Ibaraki, Japan
shiroy@iplab.cs.tsukuba.ac.jp

Kazuo Misue
Department of Computer Science
University of Tsukuba
Ibaraki, Japan
misue@cs.tsukuba.ac.jp

Jiro Tanaka
Department of Computer Science
University of Tsukuba
Ibaraki, Japan
jiro@cs.tsukuba.ac.jp

Abstract—In recent years, music-finding services have been increasing. If we have explicit information specifying pieces of music, we can find music to our taste using such services. This paper describes a tool to support music discovery. The tool visualizes a relational structure among music genres and the music-preference data of many listeners to make the users aware of their favorite music without explicit information. A case study is described to illustrate the usefulness of the tool.

Keywords—preference; music genre; sub-genre; relational structure

I. INTRODUCTION

In recent years, music distribution on the Internet has been actively developed, and it has become increasingly easy for many listeners to obtain music. We can search for and buy songs at online shops like the iTunes store¹ and Mora². Song titles and artists' names of songs are useful clues to finding the desired songs. If we have such information for our favorite songs, we can quickly find and obtain them using the services of online shops because these shops offer a huge database of music information.

We believe that there are many songs that we would like, but we do not have the necessary clues for them; we want to find songs without having these clues.

We propose a means of support for finding preferred songs without explicit clues. We focus on music genres and listeners' preferences. Music genres are groups of songs classified by their features[2]. Listeners' preferences are useful in filtering favorite songs from a huge collection of songs[1]. We visualize a relational structure among genres and sub-genres of music and listeners' music preferences. We have developed a tool to support the finding of favorite songs by visualizing these relational structures. The tool does not explicitly recommend songs, but rather provides an environment where users become aware of their taste in music.

In this paper, we describe a tool to support music discovery using music-preference data. First, we explain music genres and music-preference data. Then, we give an overview of our tool and operation of the tool, and

describe a method for drawing relational structures. Finally, we describe our case study and offer conclusions.

II. RELATED WORK

A. Using Music Genres

Yaxi et al. focused on customer-defined music genre tags defined by users of Last.fm³[3]. They visualized collaborative tags using Euler diagrams. The users can search songs by traversing the genre structure hierarchically. Their tool gives readable visualizations when the number of genres is small. We allowed listeners to look for their favorite songs from a wider range by showing larger relational structures with more music genres and sub-genres.

B. Using Listeners' Music Preferences

Takekawa et al. combined listeners' preferences and features of music such as rhythm and tempo [4]. This combination gives other clues for music searching. They used MIDI data as the music data, so it is easy to obtain music features and combine them with users' preferences. However, their method covers a very limited number of songs. Ito et al. analyzed chord progressions of music and divided them into clusters [5]. They enable the use of chord progressions as clues to finding favorite songs. However, the users need knowledge of the music theory related to chord progression. We developed a tool that allows users to search their favorite songs without expert knowledge of music.

C. Using Artists

Sarmiento et al. developed a visualization tool to show networks consisting of similarities of artists and user-defined tags in the data set of Last.fm[6]. They provided the users with high-precision browsing of these networks. We provide higher-precision browsing to show the relational structure among genres using music-preference data.

III. CLUES TO FINDING FAVORITE SONGS

A. Structures of Music Genres

Music genres are categories of songs classified by their musical features. Examples of genres are *pop*, *rock* and *folk*.

¹<http://www.apple.com/jp/itunes/whats-on/>

²<http://mora.jp/>

³<http://www.lastfm.jp>

Table I
A CONCRETE EXAMPLE OF MUSIC-PREFERENCE DATA

Listener	Song	Artist	Genre	Sub-genre	Music information	Record
white_luc	AM to PM	Christina Milian	R&B	dance	http://bit.ly/eAR4uC	2010/12/01
pommedepin	Misty	HitchcockGoHome!	folk	post-rock	http://bit.ly/byeNnZ	2010/12/01
tedman1990	Champagne	CAVO	rock	alternative rock	http://bit.ly/1QeGW	2010/12/02

Music genres may have sub-genres. Sub-genres are more detailed categories of songs. For example, *rock* has sub-genres such as *hard rock* and *progressive rock*. Some genres may share sub-genres; this means that a genre structure is not a tree. For example, the genre *pop* includes a sub-genre *pop punk* and *punk* also includes this sub-genre. Some songs may belong to some genres, but not belong to any sub-genres.

The relational structure among music genres work as a guide to finding favorite songs. We aim to support searching of listeners' favorite music by visualizing the relational structure among music genres.

B. Listeners' Preferences

Many listeners tweet and share their favorite songs using online services; one of the best-known services is Last.fm. By collecting such tweets, we can observe listeners' music preference. The preferences data includes the listener's name, song title, artist's name, genre, sub-genre, timestamp, and other information (e.g., the artist's official website and a site that introduces the song). Table 1 shows a concrete example of the data.

We considered using the preference of users with the same properties, for example, of the same generation. This is because we think people with the same properties have similar preferences. Here, we focus on generation as the users' property. We collected preference data of listeners of the same generation to extract the relational structure among genres.

C. Visualizing Structure of Genres and Listeners' Preferences

A genre structure constructed using listeners' preference data may be useful for finding favorite songs, but the structure is too complicated for many listeners to understand. Therefore, visualizing the structure helps listeners to identify the relationships between genres and sub-genres and guides them in finding their favorite songs. The relational structure among genres is a graph. We adopted a node-link diagram to show the relationships between genres and sub-genres.

We can observe which genres are much listened to by analyzing many listeners' music preferences. We therefore visualize which genres are listened to more or less by using size and the opacity of the colors of elements in the diagrams. This allows users to intuitively obtain preference information for the same generation. The users may find their favorite songs by referring to genres and sub-genres,

and comparing their preference with those of other users. Furthermore, we want to give listeners an 'awareness' of their own preferences and show the differences among listeners' preferences.

IV. TOOL TO SUPPORT FINDING FAVORITE MUSIC

We developed a tool to support music discovery based on the discussion in the previous section. We visually represent listeners' preference data to help find users' favorite songs.

A. Overview of the Tool

Figure 1 shows the initial screen of the tool. Before starting to use the tool, users have to specify their generation and preference data. Here, we introduce examples of the screen using the first author of this paper, who is in her twenties.

In Figure 1, the blue filled circle in the right-hand side of the window represents listeners. We call the circle the listeners' node. The node represents the set of listeners in the same generation; in our example, we have listeners in their twenties. The circles in the middle column of the window represent genres. We call them genre nodes. The text labels in the left-hand side of the window represent sub-genres. We call them sub-genre labels. We connect the listeners node and the genre nodes, and the genre nodes and the sub-genre labels with curved lines. We call the curved lines links. The size of a genre node indicates how often songs belonging to the genre are listened to by other listeners. For example, in Figure 1, *rock* at the top of the screen, is the biggest circle. Therefore, we understand that listeners in their twenties listen to rock music a lot.

The white lines surrounding the listeners node, genre nodes and links, and highlighting of the sub-genre labels with pink, represent the user's own preferences. We can compare our favorite genres and sub-genres with other listeners in the same generation by looking at the white lines and highlighted labels.

Genres have different colors. It is therefore easy to observe connecting links by their different colors. We understand that sub-genres connect with many genres. For example, in Figure 2, *indie* (the second top) is a sub-genre connected to four genres. We find a sub-genre connected to many genres by different colors of genres. Links have opacity. In this way, the tool shows how to listen to songs belonging to a genre related to a sub-genre for other listeners. For example, in Figure 2, comparing the link connecting *rock*

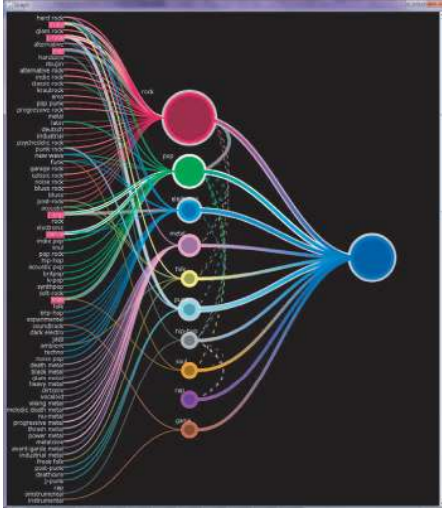


Figure 1. The initial screen of the tool.

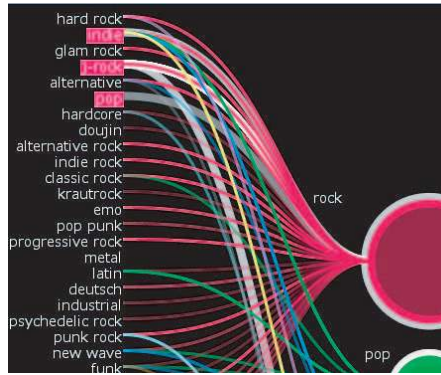


Figure 2. Links connecting genres and sub-genres.

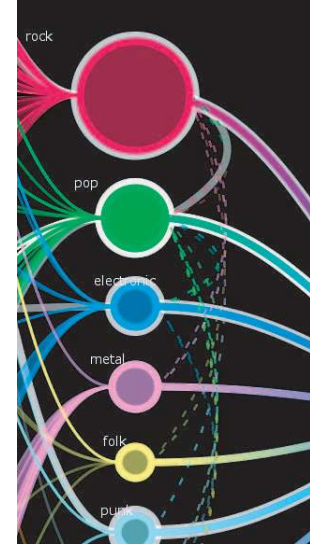


Figure 3. Links connecting two genres.

and *krautrock* (twelfth from the top) and the link connecting *rock* and *emo* (thirteenth from the top), we find that the latter link has a deeper opacity than the former. The tool shows that a link has many songs if the link opacity is deep, and the link has few songs if the link opacity is paler. In this way, we can compare the number of songs in sub-genres by the opacity differences.

Some songs belong to many genres but do not belong to any sub-genres. In this case, the tool shows a pair of genres connected through a dotted line. We therefore understand the relationship between two genres if an artist belongs to some genres but not to any sub-genres. Figure 3 is an extended illustration of links connecting a pair of genres. The links connecting a pair of genres have opacity, the same as links connecting a genre and a sub-genre. From the differences in the opacities, we can compare the number of songs belonging to specific genres.

B. Operation of the Tool

The tool displays artists' names by clicking a genre node or a sub-genre label. The artists' names shown represent other listeners' favorite artists. We therefore consider that it is easy for users to search unknown artists. We show an example of clicking a genre node, in Figure 4, and clicking a sub-genre label, in Figure 5. These allow users to find their favorite artists from a list of artists' names.

1) *Finding Many Unknown Artists*: The tool shows the names of many artist listened to by many listeners. Therefore, if we search for unknown artists, we can find them by focusing on artists not highlighted in white or pink.

2) *Finding Many Artists in User's Own Favorite Genre*: We can find many artists' names in a genre because genres have many sub-genres. For example, if users focus on their

favorite genres, they know that they listen to two or three sub-genres, but the genres have many sub-genres. Therefore, we can search artists from unknown sub-genres. Focusing on a genre highlighted in white, and sub-genres related to genres not highlighted in pink, allows users to find many artists.

3) *Finding Many Artists Focusing on Genres Relationships*: As mentioned in Section III-A, a genre has many sub-genres, and a sub-genre belongs to a number of genres. Some songs do not belong to any sub-genres, but belong to a number of genres. A genre and a sub-genre, and a pair of genres, are connected by links showing relationships, and genres have colors. The tool therefore allows users to find many artists by traversing links, comparing the differences in link colors.

V. DEVELOPMENT OF THE TOOL

In this section, we first describe the music-preference data used in our research. Next, regarding the visual presentation of the used data, we show how to define the size of genre nodes and how to draw links.

A. About Music-preference Data

Last.fm is a popular music social network service. Last.fm synchronizes with iTunes⁴ or Windows Media Player⁵, and gathers data on music listened to by users. The service also shows listeners who liked users' preferences for music and artists whom users may like. The service allows users to register their favorite songs by clicking the "Love track" button.

⁴<http://www.apple.com/jp/itunes/what-is/>

⁵<http://windows.microsoft.com/ja-JP/windows/products/windows-media-player>

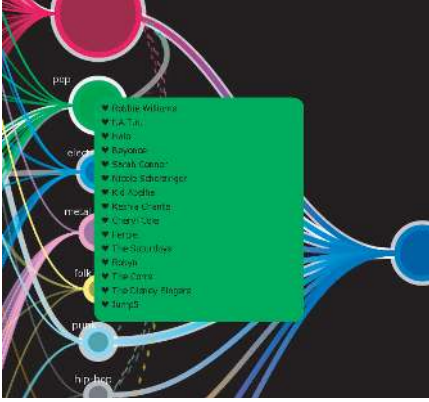


Figure 4. Clicking a genre node *pop*.



Figure 5. Clicking a sub-genre label *classic rock*.

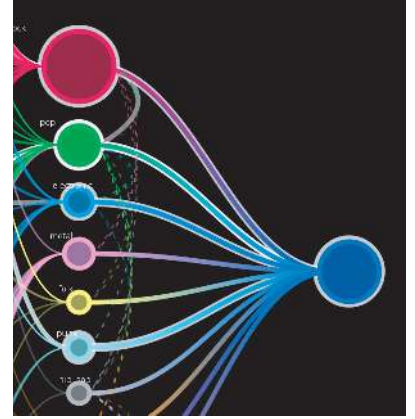


Figure 6. Link connecting genre nodes and listener node.

If the users set up a collaboration service on Twitter⁶ and Last.fm, they can tweet automatically about their favorite songs adding “#lastfm”, that is, a hash-tag.

We gathered tweets with the hash-tag “#lastfm”, using the Twitter searchAPI⁷. The gathered tweets data includes user name, song title, artist’s name, URL where the song information appears, and time-stamp. Information about genres and sub-genres was obtained manually from customer-defined tags of Last.fm. We used the gathered tweets data and the genres and the sub-genres obtained from customer-defined tags of Last.fm as listeners’ preference data.

We extracted the genres and the relationship among genres from listeners’ preference data. The total set of songs is $M = \{m_1, m_2, m_3, \dots, m_n\}$. The set of extracted genres is $G = \{g_1, g_2, g_3, \dots, g_k\}$. The set of sub-genres is $S = \{s_1, s_2, s_3, \dots, s_l\}$. Genre $g_i \subset M, (i = 1, 2, \dots, k)$ is a subset of songs, and sub-genre $s_j \subset G, (j = 1, 2, \dots, l)$ is a subset of genres.

B. Deciding the Radius of a Genre’s Node

The radius of a genre’s node is decided by the ratio of the number of songs contained in a genre to the total number of songs M . Expression (1) shows how to obtain the radius of the node that represents the intended genre $g_i (i = 1, 2, \dots, k)$:

$$r_i = c_1 \cdot \frac{|g_i|}{n} + c_2 \quad (1)$$

Where c_1 and c_2 are constants, and n is the total number of songs. For some genres, the ratio becomes very small, regardless of the number of songs. We therefore set a minimum radius of c_2 to ensure that all genre nodes can be seen.

⁶<http://twitter.com/>

⁷<http://apiwiki.twitter.com/w/page/22554679/Twitter-API-Documentation>

C. Links Using Curved Lines and Color Combination of Links

1) *Using Curved Lines and Making Links:* We connect the listeners node and genre nodes, and genre nodes and sub-genre labels by links of curved lines. Using curved lines allows bundling of some links and decreases the hiding of links. By focusing on connecting of nodes or labels, we can guess the number of songs from crowding of the links. We have been inspired by Holten’s edge bundles[7].

2) *Color Combination of Links:* We assign opacity to links connecting genre nodes and sub-genre labels. The opacity shows how many songs a sub-genre contains. We show how to set up the opacity α by expression (2):

$$\alpha_j = c_3 \cdot \frac{|s_j|}{n} + c_4 \quad (2)$$

Where c_3 and c_4 are constants, and n is the total number of songs; $\alpha = 0$ indicates complete transparency and $\alpha = 1$ indicates complete opacity. For some sub-genres, the ratio becomes very small, regardless of the number of songs. We therefore set c_4 as the minimum.

Links connecting the listeners node and genre nodes use gradations of each genre color and each listener’s color. If connected links use only color, there may be inconsistencies between the nodes and the root parts of the link connected to the nodes when some nodes use a number of colors. In order to avoid this, we use gradation and we use the same color for the node and the link connected to the node. Figure 6 shows an enlarged view of the link connecting the listeners node and genre nodes.

VI. CASE STUDIES

To verify the availability of our tool, we performed case studies.

Table II
RESPONSES OF EACH USER IN EACH PATTERN

Pattern	User A	User B
Pattern 1	Pendulum (electronic) Papa (electronic) Muse (alternative) Mellodrone (indie rock)	McFly (pop rock) Hurts (synthpop)
Pattern 2	not discovery	kagraa (j-rock) Supercar (j-rock) Omar A.Rodriguez-Lopez (progressive rock) Bullet For My Valentine (metalcore)
Pattern 3	Fort Minor (hip-hop) Handsome Boy Modeling School (hip-hop) Bad Religion (punk)	Sparks (new wave) anNina (j-pop)

A. Users

We asked a graduate student and an undergraduate student, who are studying information technology, to use our tool. They listen to music in their daily life. We prepared their music-preference data. User A likes the genres *rock*, *pop*, and *folk*, and the sub-genres *j-rock*, *punk rock*, *j-pop*, and *indie*. User B likes the genres *rock*, *electronic*, and *metal*, and the sub-genres *j-rock*, *progressive rock*, *new wave*, *post-rock*, *j-pop*, and *metalcore*.

B. Setup

We gave the users a five-minute explanation of how to use our tool, and then we asked them to use our tool. We showed the users a screen displaying highlights of their preferences. We asked them to search artists using the three patterns below, and to preview artists' songs on YouTube⁸.

Pattern 1

Search artists from genres or sub-genres of listeners with similar tastes.

Pattern 2

Search unknown artists from their favorite genres or sub-genres, and find songs.

Pattern 3

Search artists from their favorite genres belonging to other genres or sub-genres, and find songs.

If users find their favorite music, we ask them to record in their response sheets the artists' names and genre names or sub-genre names that they found. However, if users do not find their favorite music, we ask them to write "not found" in their response sheets.

Next, we asked users to search artists freely. We then asked them to describe in their response sheets how to search and find artists. Finally, we asked them to write freely about their overall impression of using our tool. We used listeners' preference data for users belonging to the same generation in our case study.

C. Results

We show the results for the three patterns in Table II. Responses are shown as "artist (genre or sub-genre)".

Regarding the three patterns, user A followed the genres he was interested in by tracing the links connecting the

⁸<http://www.youtube.com/>

Table III
RESPONSES OF EACH USER WHEN THEY SEARCH FREELY

Pattern	Artist	Genre or Sub-genre
Pattern A	Dr. Dog	indie
Pattern B	The Corrs	folk
Pattern C	Hocico My Chemical Romance Emil Bulls	dark electro emo nu-metal

nodes, using the mouse cursor. He specifically searched *electronic* and *hip-hop* in Pattern 1 and Pattern 3. In Pattern 3, he discovered two *hip-hop* artists. However, overall, he did not follow sub-genres very much.

User B followed the genres and sub-genres she was interested in by tracing the links connecting the nodes and labels, using the mouse cursor. She specifically searched *pop*, and genres and sub-genres related to *pop* in Pattern 1 and Pattern 3. For example, she found all artists belonging to sub-genres of *pop* in Pattern 1 and Pattern 3. User B focused on sub-genres rather than on genres.

Both users followed the colors of the links because genres have colors. However, when they followed sub-genres, they seemed to lose their way. When they reached genres or sub-genres of interest, they previewed some of the artists instead of all the artists. We interviewed them about how to search artists. They commented> "When searching artists, first I searched artists whose names I have heard of. Next, I searched artists' names that interested me."

Next, we asked the users to search artists freely. The users searched artists in the following patterns.

Pattern 4

Search artists in sub-genres that connect most genres.

Pattern 5

Search artists in genres that they listen to a lot, but that other listeners do not listen to very much.

Pattern 6

Search artists in sub-genres that belong in their favorite genres.

Pattern 4 and Pattern 5 are how user A searches for artists, and Pattern 6 is how user B searches for artists. In Table III, we show the results for each pattern.

D. Discussion

Both users used our tool in a similar way in Pattern 1 to Pattern 3. However, we obtained different results from user A and user B. User A mainly searched artists from genres he listened to a little. User B searched artists from sub-genres similar to her favorite genres. The focus of each user is different. However, in Pattern 1, they could search music they like from genres or sub-genres that they do not usually listen to because they focused on what other listeners listen to. As in the case of Pattern 1, we believe that in Pattern

3 they could search their tastes in artists from the genres or the sub-genres because they focused on the relational structure among the genres. Color-coding of genres and curved lines connecting genre nodes and sub-genre labels support the understanding of the relationships among genres. We therefore believe that our tool supports searching a wide range of songs.

In Pattern 2, user B could find the artists she likes; however, user A could not. This is because the listeners' preference data we used included a lot of Western music, rather than Japanese music. Both users commented: "We want more data with Japanese music." We had gathered the data by combining the services of Last.fm and Twitter, and Last.fm is more widely used by overseas users. Therefore, overall, there were few Japanese music data. However, with Pattern 2, it is possible to find unknown artists in the user's favorite genres or sub-genres.

When we asked users to search artists freely, we saw that they searched in a different way than we had expected.

In Pattern 4, user A searched by focusing on the genre relationships, regardless of his favorite genres and sub-genres. We obtained the following comment from user A: "I found a sub-genre contained many genres by color-coding of genres. Therefore, I got interested and searched artists." The color-coding of genres is therefore considered to be effective in showing the relational structure among genres.

In Pattern 5, we found that the user searched artists based on the differences between the other users' preferences and their own preferences. Our tool shows the user's music-preference data by highlighting in white. The tool shows other listeners' music-preference data by other colors. The opacity of each link shows how many songs sub-genres contain. We obtained the following comment from user A: "I could look at my music preference objectively by comparing my music preferences with those of others." Therefore, using colors and opacity is considered effective in comparing a user's music preference with others.

Pattern 6 shows how to search artists in unknown sub-genres of personally preferred genres. A genre has many sub-genres. Comparing one's music preferences with sub-genres, it is often the case that there are unknown sub-genres. By showing these sub-genres, users can search a lot of music in their favorite genre.

VII. CONCLUSIONS

We developed a tool to support listeners' tastes in music by showing listeners' music preference, and the relationships between genres, by visualizing the listeners' preference data.

Through our case study, we made it possible to identify users' tastes in music. By showing a user's preference data and those of others at the same time, we enabled users to look at their music preferences objectively.

ACKNOWLEDGMENTS

The authors would like to thank all the case study participants, and Dr. Simona Vasilache and the anonymous reviewers for their valuable comments.

REFERENCES

- [1] Yoshinori Hijikata, "User Profiling Technique for Information Recommendation and Information Filtering", *Transaction of the Japanese Society for Artificial Intelligence vol.19, no.3*, pp.365-372, 2004 (in Japanese).
- [2] Satoshi Masuda, "What is Musical Genre?: The Topology of Sounds, Concepts, and Power", *10+1 No.28*, pp.27-30, 2002 (in Japanese).
- [3] Yaxi Chen, Rodrigo Santamaria, Andreas Butz, Roberto Theron, "TagClusters: Semantic Aggregation of Collaborative Tags beyond TagClouds", *Proceedings of 10th International Symposium on Smart Graphics, Springer Press*, pp. 56-67, 2009.
- [4] Kazuki Takekawa, Yoshinori Hijikata, Shogo Nishida, "Implementation of a Content-based Music Search and Recommendation Method", *DEWS2007 (The Institute of Electronics, Information and Communication Engineers, 18th, Data Engineering Workshop)*, pp.1-8, 2007 (in Japanese).
- [5] Takayuki Ito, Reiko Miyazaki, Misuho Oda, Makiko Nagasawa, Chiemi Watanabe, "Visualization of Musical Information Using "HeiankyoView"", *IPSJ SIG technical reports 2008 (50)*, pp.71-76, 2008 (in Japanese).
- [6] Luis Sarmiento, Fabien Gouyon, Bruno G. Costa, "Visualizing Networks of Music Artists with RAMA", *International Conference on Web Information Systems and Technologies (WEBIST 2009)*, pp.232-237, 2009.
- [7] Danny Holten, "Hierarchical Edge Bundles Visualization of Adjacency Relations in Hierarchical Data", *IEEE Transactions on Visualization and Computer Graphics 2006*, pp.741-748, 2006.