Tonic: Combining Ranking and Clustering Dynamics for Music Discovery

Dimitrios Bountouridis
Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
d.bountouridis@uu.nl

Jan Van Balen
Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
j.m.h.vanbalen@uu.nl

Marcelo Rodríguez-López
Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
me.rodriguezlopez@uu.nl

Anna Aljanaki
Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
a.aljanaki@uu.nl

Frans Wiering
Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
f.wiering@uu.nl

Remco C. Veltkamp
Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
r.c.veltkamp@uu.nl

ABSTRACT

This paper describes the design of TONIC, a novel web interface for music discovery and playlist creation. TONIC maps songs into a two dimensional space using a combination of free tags, metadata, and audio-derived features. Search results are presented in this two dimensional space using a combination of clustering and ranking visualization strategies. TONIC was ranked first in the 2014 MIREX User Experience Grand Challenge, where it was evaluated in terms of learnability, robustness and overall user satisfaction, amongst others.

Categories and Subject Descriptors

H.5.m. [**Information Interfaces and Presentation (e.g. HCI)**]: Miscellaneous

General Terms

Design

Keywords

user-experience, music discovery, HCI, music information retrieval

1. INTRODUCTION

Owning physical copies of music, such as CDs and vinyl records, seems to be slowly becoming a pastime exclusive to audiophiles and, some might even say, an anachronism. Conversely, online streaming services give to their users access to millions of songs and have become the standard for casual listening. This cheap and fast access to vast music collections has renewed interest in music discovery: the task of efficiently finding unknown music that is interesting to the user, either for their personal enjoyment or other activities, such as selecting music for an event or a video.

While plenty of systems have been proposed to aid music discovery, both in the Music Information Retrieval (MIR) domain and in industry, the problem is still of interest. In terms of recommendation strategies, two common obstacles are (a) the so called "long tail" problem, which refers to the fact that most songs in a collection will have little or unreliable meta-data and user data, and (b) the "cold start" problem, which states that typical systems based on user behaviour (e.g. collaborative filtering) have difficulties dealing with unseen data [2].

In terms of interfaces, it has proven difficult to create an interface that offers a completely satisfactory user experience, providing high usability, engagement, good information organization, all while promoting discovery. Music itself is inherently complex and multidimensional, with relationships between songs, artists and record labels. This makes it hard to compactly present, in an appropriate manner, the information to the user.

In this paper we describe TONIC[1,2], an interface for music discovery that aims to fuse two strategies for visual organization: ranking and clustering.

The rest of the paper is organised as follows. In section 2 we discuss related work, in section 3 we introduce TONIC's design, in section 4 we describe its implementation, and finally, in section 5, we discuss its evaluation.

AM '15 Thessaloniki, Greece

http://dx.doi.org/10.1145/2814895.2814930

2. RELATED WORK

Music discovery and music discovery interfaces relate to the fields of Music Information Retrieval (MIR) and visualization, in which a good amount of research has been

[1] Available at www.projects.science.uu.nl/COGITCH/Tonic
[2] Video demo at www.youtube.com/watch?v=ucSzaLrzDvM

conducted over the past years. One of the first content-based MIR visualizations was *Islands of Music* [8] which uses self-organizing maps to project songs on a two-dimensional space, and represents clusters of similar songs as "islands" in that space. Variations of the same concept were presented by Schedl *et al.* [10] and Mörchen *et al.* [7]. Knees *et al.* [4] presented a similar method but on a freely-navigated three-dimensional space where the height of "islands" corresponded to the similarity between the cluster of songs. A three-dimensional approach by Lamere *et al.* [5] called *Search inside the music* used audio-content similarity to project similar songs adjacently in the space. Torrens *et al.* [11] investigated three different visualization techniques (disc, rectangle and tree maps) for exploring music libraries but found no particular standout, since each one of them offered its pros and cons.

Aurally aided visualizations for music exploration were investigated by Lübbers [6] and Brazil and Fernstrom [3]. Schedl, in addition, has investigated a great amount of visualization techniques for hierarchical Web music content, from treemaps to 2D and 3D sunbursts [9]. Explaining the details of each method exceeds the scope of this paper.

Lately, a number of mobile Apps have emerged aiming to music discovery via elegant visualization. *Discovr*[3], which is based on The Echo Nest[4], presents to the user a collapsible force-layout artist similarity network. Starting from an artist name as the root, the user can explore and expand different branches. A similar approach, but for desktop computers, is *MusicExplorerFX*[5]. *Hitlantis*[6] is a 2D interface with artists presented as planets, colour coded according to genre, orbiting around the center. The closer to center the more popular the artist, while the larger the planet the more songs have been uploaded for that artist. Both Apps work on the artist level and while *Discovr* does not support any form of clustering, *Hitlantis* provides only genre clustering.

Despite the proliferate research on MIR visualization and the success of mobile Apps, commercial music services have not adopted any of the aforementioned approaches. In contrast, commercial services typically use a list-like representation of relevant songs, similar to result organization of Web search engines. Spotify[7], Tidal[8] and Deezer[9] use curated playlists as their main structure for music discovery. Songs are typically grouped into subjective categories such as "Top of The Morning", "Dinner Time", and so on, in addition to genres, moods, decades and so on. Although each category can be multi-layered (e.g. different kinds of rock music), the whole set of songs included is limited and probably hand-picked. Moreover, playlists are presented in static list-format while categories cannot be combined (e.g. search for soul music in the 80's decade). The user can discover new music only via exploring the artist similarity network or, in the case of Spotify, by following the user-specific suggestions derived from his listening habits. Beats Music[10] suggests new music to the user after they have completed a favourite genre and artist selection process. In terms of visualization, genres and artists appear as circles in a topology that roughly preserves the similarities between them (e.g. "dance" located next to"R&B"). However, beyond that point, the system reverts to the typical list-format representation.

Tonic aims to revamp spatial layouts for music discovery, by focusing on a use case in which the music collection is mostly unknown – as opposed to the use case of commercial services, in which users expect to see familiar songs – and focusing on interactions. Tonic was designed for the MIREX 2014 User Experience Grand Challenge (GC14UX), a challenge in which each participant competes to design a complete, user-facing music information retrieval system. As part of GC14UX, all participating systems were subjected to an extensive user-centered evaluation of the user experience.[11]

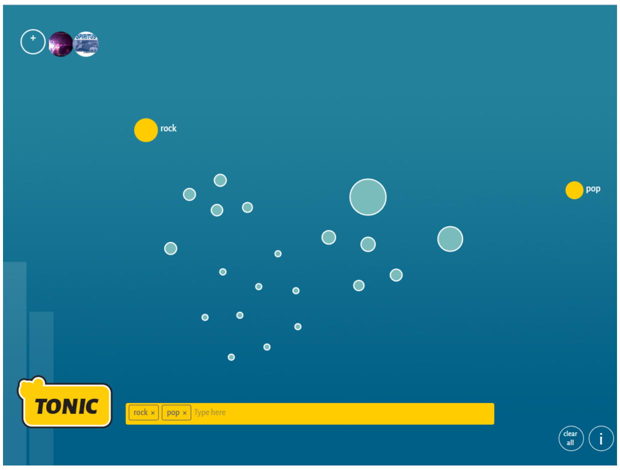

Figure 1: A screenshot of Tonic.

3. DESIGN

Tonic was designed taking into account three primary principles, which manifest themselves in three layers: the data layer, the interactions with the interface in which the data are displayed, and the interactions with the data. The design for each of these layers derives from the choices in the underlying layer. We therefore begin with the music collection.

3.1 Collection of Little-known Music

The music collection for which Tonic was built is a large collection of little-known songs. The songs are not necessarily grouped in albums, but they are annotated with several kinds of tags or keywords. These can be manually provided or automatically extracted.

The particular collection that was used in the first version of Tonic, was a 10,000 song subset of the Jamendo dataset (see Section 4.1). The collection contains a variety of structured and unstructured keywords, including information about genre, instrumentation, geographic origin and language. The processing of these tags is also explained in detail in Section 4.1.

[3]itunes.apple.com/us/app/discovr-discover-new-music/id412768094
[4]the.echonest.com
[5]musicexplorerfx.citytechinc.com
[6]www.hitlantis.com
[7]www.spotify.com
[8]tidal.com
[9]www.deezer.com
[10]www.beatsmusic.com
[11] www.music-ir.org/mirex/wiki/2014:GC14UX

To these metadata provided by Jamendo, a number of automatically extracted features were added. This includes information about tempo and mood, among other things.

The consequences of the nature of this collection for the system's design are twofold. Since most of the music is little-known, artists and titles are typically unknown to the user and therefore not useful as queries in standard meta-data search. Users simply wouldn't know what to look for. However, a relatively rich set of labels is available to be used as keywords instead. Moving towards the interface layer, this also has implications for the layout: songs need not be presented in a standard title-artist-album table layout as they are usually displayed in list-like interfaces.

As a first important design decision, the interface layer itself is therefore characterized by a representation of songs as unlabeled geometric objects floating around in a plane, as shown in Figure 1– the light-blue bubbles are songs.

3.2 Interface Interactions

Crucially, in TONIC, all search keywords (tags, genre, mood...) entered as a query by the user, are displayed among the results (songs) in the same 2D space. The tags can then be moved around in the space. This allows TONIC to combine two kinds of dynamics in the organization of the space: ranking, and clustering.

Clustering is achieved by making each of the keywords attract the songs that satisfy it. In the simplest case of two query keywords A and B, this will generally create a clusters with results that satisfy A, a cluster that satisfies B, and a cluster between the two that satisfies both. (Songs that satisfy neither A or B will not be displayed.) The space separating the clusters relating to two tags can be manipulated by separating the respective tags, a very intuitive interaction that helps the user organize the results space.

Figure 2: A screenshot of the automatic audio player of Tonic. The tags of the song currently playing are also presented and color-coded according to their weights; higher-weight tags appear brighter.

Ranking is achieved by introducing a constant pull from below on all results. This makes the absolute height of each song object an indication of its combined relevance with respect to all keywords. This results in a vertical ranking of the songs. To emphasize this ordering, song objects increase in size as they move up in the results space, giving an intuition of results "bubbling up". More importantly however, this additional dynamic creates an intuitive way for the user to give a weight to each keyword in their query: by moving the keyword up and down. Moving keyword up will make each matching song follow it to the top, while moving a keyword down will make each of its matching songs sink back to the bottom.

More details of the ranking and clustering dynamics, with a diagram, are given in Section 4.4.

3.3 Data Interactions

A final important design choice involves the user's "search cycle", the loop in which a user iteratively specifies their keywords, inspects the results, optionally saves some results, adjusts keywords, etc. several times in a row, until the search is over (either because enough results were found or because the process is found to not lead anywhere better).

We chose to make the search cycle as simple and short as possible. To this end, we concentrate most of the interaction in the "inspection" step, in *listening*: bubbles that are clicked will start playing. The artist and title will also show up, but are moved to a bar at the bottom to keep the feedback from listening dominant.

In the last step, we facilitate the adjustment of keywords by making it possible for the user to click keywords belonging to a song (displayed next to title and artist), to add them to the list of query keywords in the search bar. This provides a simple "give me more of this" mechanism. It is equally easy to remove queries from the search bar (by clicking a little "X"), or remove all query keywords altogether.

3.4 Additional Design Features

At the end of each successful search cycle, interesting results can be stored by dragging them onto a "plus" button in the top left corner. Songs can be removed again form this tray by dragging them back onto the results field. Though this had not been implemented yet, a future version of TONIC might see the stored songs direct to www.jamendo.com, so that, at the end of the search, results can be downloaded or bookmarked in the user's browser.

Finally, all these features are explained in a short walk-through at the beginning of a user's first session. A set of help pointers float above the interface, while everything is functional as usual. Throughout, visual feedback in the form of slow blinking is used to indicate where sounds come from (generally, a song that is being previewed).

To sum up, we propose an interactive music search framework that is based on keywords and tags rather than meta data. The user interface presents both the query and the results in a shared 2D space that combines interactive ranking and clustering dynamics. Meanwhile, the user experience is enhanced by simplifying the flow of feedback (listening and making adjustments to the query) in the search cycle.

4. IMPLEMENTATION

4.1 Dataset & Features

We use a 10,000 song subset of the Jamendo dataset as provided by the GC14UX challenge. Jamendo[12] contains mostly music of unsigned, and therefore typically unknown artists of various genres and moods. Beside the typical metadata (e.g. artist, song title, duration etc.) each song comes with information regarding the language of the lyrics, the speed of the song, whether it is acoustic or electric, the instruments used, the gender of the artist and others. The values for these categories are typically derived from a categorical fixed set. In addition, each song comes with a list of free tags accompanied by counts as provided by listeners (e.g. rock:8, pop:3, vocal:1).

We incorporate content-based mood tags by applying music emotion recognition method to the audio using a method described in [1]. We also use Essentia[13] toolbox to add descriptive tags related to loudness, tempo, timbre. We use

[12] www.jamendo.com

[13] essentia.upf.edu

automatic tempo estimation and onset estimation algorithm from Essentia to predict the tempo, and Gaia model for brightness of timbre.

For our system, fixed categories, free tags and moods were merged into a set of tags for which relative weights were assigned. Although assigning and normalizing weights can be performed in various ways, we speculate that the difference between them is negligible in our context. The reason is twofold: firstly, listeners have a fuzzy and subjective perception of the categories that a song belongs to. Secondly, the way that TONIC places songs in the two-dimensional space and its discovery-driven nature, allows for flexibility in the overall ranking.

4.2 Back-end & Front-end

All metadata are stored in a MySQL database, and all audio (60GB) is hosted in our University's servers. The front-end is *D3.js* based interface that uses *Soundmanager.js* for streaming the audio files. *D3.js* is a cross-platform framework that provides powerful visualization modules that are dynamic, interactive and can be easily adapted. Its power derives from the ability to bind Document Object Module (DOM) elements to data and then applying data driven transformations to the document. In addition, *D3.js* offers physics-based visualization modules such as forced layouts, graphs and bubble charts, making it an ideal solution for our design's needs. At its foundation, TONIC is simply a bubble chart with multiple gravity forces that attract bubble objects at different strengths. As a consequence, its main functionality can be reproduced easily by extending on some bubble chart examples of *D3.js*.

4.3 Query Pipeline

Each song s_i in the database is represented by a set of tags and weights $T_i = \{t_i^1, t_i^2, ..., t_i^n\}$ and $W_i = \{w_i^1, w_i^2, ..., w_i^n\}$. Each user query is a set of keywords $Q = \{q_1, q_2, ..., q_m\}$ where q_1 is the newest query keyword. As soon as the query reaches the server, our systems tries to find all songs that satisfy the following expression: $q_1 \cdot (q_2 \cdot q_3 \cdot ... \cdot q_m)$. Therefore, we try to find all possible songs that satisfy each and one of the query keywords. As the size of Q grows larger or keywords become more particular, the number of matches that are returned can be close to zero, even for a set of 10,000 songs. In order to avoid such a development we gradually relax the constraints in the expression by introducing random OR clauses inside the parentheses, until we get at least 10 songs. For example:

$$1^{st} \text{ attempt: } q_1 \cdot (q_2 + q_3 \cdot ... \cdot q_m)$$
$$2^{nd} \text{ attempt: } q_1 \cdot (q_2 + q_3 \cdot ... \cdot q_k + q_{k+1} ... \cdot q_m)$$
$$...$$
$$m-2 \text{ attempt: } q_1 \cdot (q_2 + q_3 + ... + q_m)$$

Our aim is to find all songs that satisfy at least q_1 and the largest number of the remaining keywords. It is possible that no songs can satisfy the query due to typos; in that case the system returns no results. In the matching scenario, the system returns 10 songs randomly selected from the results list, ensuring that they have not been returned in a previous query.

4.4 Physics

As soon as the relevant songs are returned to TONIC's front-end, their corresponding bubbles are created and placed randomly in the 2D space. Their position is updated until they reach an equilibrium through a dynamic physics system.

The keyword bubbles act not only as weight holders but as gravitational forces. Each song bubble is pulled towards a keyword with a strength relative to the matching tag's weight. The overall attraction force r_i, which defines the overall relevance of the song with regard to the keywords and their weights, is computed as:

$$r_i = \sum_{j \in T_i \cap Q} w_i^j v_j \frac{|T_i \cap Q|}{|Q|} \quad (1)$$

where $V = \{v1, v2, ..., v_m\}$ are the keyword weights corresponding to Q and $|T_i \cap Q|$ is the number of query keywords satisfied by the song. These forces of attraction can be decomposed to their vertical and horizontal components. A function of all the vertical forces pushes song bubbles upwards in the plane and can be considered the upthrust (buoyancy) following a fluid physics metaphor.

Similar to the upthrust, the sum of the horizontal components defines the horizontal position of song bubbles (see Figure 3). The default placement of the keywords above the songs and the wider x axis, allows the users to achieve a form of spatial grouping by moving keywords along the horizontal plane. Consequently, the upthrust and sum of horizontal forces achieve two separate types of organization, ranking and clustering respectively (see Figure 4).

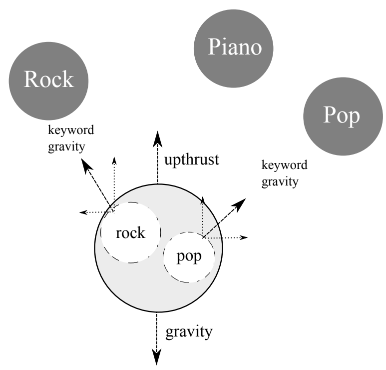

Figure 3: The forces applied to a song with two tags (pop,rock) in a system of three gravitational keyword forces (rock,piano,pop). The keyword forces can be decomposed to vertical and horizontal components. The vertical components create the upthrust while the horizontal define the song's position in the x axis.

5. EVALUATION

The goal of MIREX User Experience Grand Challenge task of 2014 (GC14UX) is to inspire the development of complete music retrieval systems and put user-experience in the center of MIR. 69 human evaluators, mostly from the MIR community, rated all submitted systems on a seven-scale rating system in terms of the following criteria: overall satisfaction, learnability, robustness, affordance (how well does the system allow you to perform what you want to do) and feedback (how well does the system communicate what

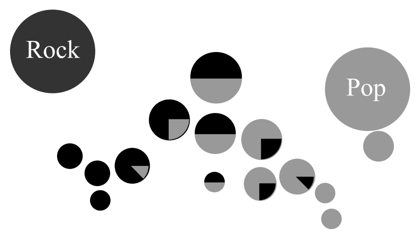

Figure 4: An example of the clustering and ranking fusion for a set of song bubbles containing the tags "rock" and "pop" at different amounts (represented as size). Songs that satisfy equally both keywords are forced in the middle between their corresponding bubbles. The rest are spaced accordingly.

Table 1: The mean and variance (in parentheses) for all three systems across the different criteria. The rating ranges from 1 to 7; $N = 69$.

	TONIC	*Moody*	*ThankYou*
Learnability	**5.35** *(2.06)*	5.27 *(2.28)*	5.33 *(2.51)*
Robustness	4.53 *(1.89)*	**4.65** *(1.98)*	4.39 *(1.82)*
Affordance	**4.79** *(2.16)*	4.68 *(2.31)*	4.27 *(2.59)*
Feedback	**4.80** *(2.63)*	4.52 *(2.68)*	4.60 *(2.37)*
Overall satisfaction	**5.17** *(2.06)*	4.68 *(2.28)*	3.97 *(2.51)*

is going on). The task given to evaluators to perform using the systems was the following:

"You are creating a short video about a memorable occasion that happened to you recently, and you need to find some (copyright-free) songs to use as background music"

The instructions are purposely vague, giving to each evaluator freedom to explore and interact with the submitted systems in different ways.

TONIC was submitted at GC14UX and was ranked first against two other systems (Moody and ThankYou). Details about the competing systems were not provided to the participants. Although a complete analysis of ratings has yet to be officially published, the results were made available online[14] (see Table 1). The overall satisfaction for TONIC and Moody were significantly higher than ThankYou with $p < 0.05$. However, no significance difference was observed between the two highest ranked systems in any of the criteria.

6. CONCLUSION

In this paper we presented TONIC, a novel music discovery interface. In TONIC the user enters a text based search query, and search results are presented visually using two intuitive and widely used organizational strategies: ranking and clustering. Songs matching the search query terms are visually represented as bubbles, which are organized in a two-dimensional space so that vertical positioning and bubble size indicate ranking, and distance between bubbles represent the degree of similarity in respect to a query term. The user can interact with the system by clicking on the bubbles to hear snippets of the represented song, and moving around clusters of bubbles to aid visualization.

TONIC was evaluated by 69 users as part of the GC14UX competition, and ranked first in respect to two other systems. The evaluation results show that TONIC excels in two evaluation criteria: learnability and overall user satisfaction.

In future work we plan to study in more depth the interaction between the ranking and clustering visualization strategies. Moreover, we aim to test the system in other music discovery scenarios, e.g. to aid the discovery of meaningful melodic motives in folk music.

[14] www.camdemy.com/media/16477

7. ACKNOWLEDGEMENTS

Marcelo Rodríguez-López (NWO-VIDI grant 276-35-001), Dimitrios Bountouridis (NWO-CATCH project 640.005.004), Jan Van Balen (NWO-CATCH project 640.005.004) are supported by the Netherlands Organization for Scientific Research. The work of Anna Aljanaki, Remco Veltkamp and Frans Wiering is supportedby the FES project "COMMIT/".

8. REFERENCES

[1] A. Aljanaki, F. Wiering, and R. Veltkamp. Computational modeling of induced emotion using gems. In *Proceedings of the Conference of the International Society for Music Information Retrieval (ISMIR)*, pages 373–378, 2014.

[2] D. Bogdanov and P. Herrera. Taking advantage of editorial metadata to recommend music. In *9th International Symposium on Computer Music Modeling and Retrieval (CMMR)*, pages 618–632, London, UK, 2012. Springer Verlag, Springer Verlag.

[3] E. Brazil and M. Fernström. Audio information browsing with the sonic browser. In *Coordinated and Multiple Views in Exploratory Visualization, 2003. Proceedings. International Conference on*, pages 26–31. IEEE, 2003.

[4] P. Knees, M. Schedl, T. Pohle, and G. Widmer. An innovative three-dimensional user interface for exploring music collections enriched. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 17–24. ACM, 2006.

[5] P. Lamere and D. Eck. Using 3d visualizations to explore and discover music. In *Proceedings of the Conference of the International Society for Music Information Retrieval (ISMIR)*, pages 173–174, 2007.

[6] D. Lübbers. Sonixplorer: Combining visualization and auralization for content-based exploration of music collections. In *Proceedings of the Conference of the International Society for Music Information Retrieval (ISMIR)*, pages 590–593. Citeseer, 2005.

[7] F. Mörchen, A. Ultsch, M. Nöcker, and C. Stamm. Visual mining in music collections. In *From Data and Information Analysis to Knowledge Engineering*, pages 724–731. Springer, 2006.

[8] E. Pampalk. *Islands of music: Analysis, organization, and visualization of music archives.* 2001.

[9] M. Schedl. *Automatically extracting, analyzing, and visualizing information on music artists from the World Wide Web.* 2008.

[10] M. Schedl, E. Pampalk, and G. Widmer. Intelligent structuring and exploration of digital music

collections. *e&i Elektrotechnik und Informationstechnik*, 122(7-8):232–237, 2005.

[11] M. Torrens, P. Hertzog, and J. L. Arcos. Visualizing and exploring personal music libraries. In *Proceedings of the Conference of the International Society for Music Information Retrieval (ISMIR)*, 2004.