geMsearch: Personalized Explorative Music Search

Christian Esswein
University of Innsbruck, Austria
christian.esswein@student.uibk.ac.at

Markus Schedl
Johannes Kepler University
Linz, Austria
markus.schedl@jku.at

Eva Zangerle
University of Innsbruck, Austria
eva.zangerle@uibk.ac.at

ABSTRACT

Due to the rise of music streaming platforms, huge collections of music are now available to users on various devices. Within these collections, users aim to find and explore songs based on certain criteria reflecting their current and context-specific preferences. Currently, users are limited to either using search facilities or relying on recommender systems that suggest suitable tracks or artists. Using search facilities requires the user to have some idea about the targeted music and to formulate a query that accurately describes this music, whereas recommender systems are traditionally geared towards long-term shifts of user preferences in contrast to ad-hoc and interactive preference elicitation. To bridge this gap, we propose geMsearch, an approach for personalized, explorative music search based on graph embedding techniques. As the ecosystem of a music collection can be represented as a heterogeneous graph containing nodes describing e.g., tracks, artists, genres or users, we employ graph embedding techniques to learn low-dimensional vector representations for all nodes within the graph. This allows for efficient approximate querying of the collection and, more importantly, for employing visualization strategies that allow the user to explore the music collection in a 3D-space.

ACM Classification Keywords

H.3.3. Information Search and Retrieval: Information filtering; H.4.2. 2. Information Systems Applications: Types of Systems: Decision Support

Author Keywords

music information retrieval, search, recommender systems, visualization, graph embedding

INTRODUCTION

In recent years, music streaming platforms have become a central means for listening to music as these allow users to access huge collections of music. This evolution has also influenced the way users search and explore music. For instance, the streaming platform Spotify currently serves 140 million active users and provides a collection of more than 30 million songs[1] (as of June 2017). Consequently, the primary objective for users has shifted from retrieving specific songs to finding and ultimately exploring songs that match certain criteria reflecting the user's current preferences and context [8, 5].

Currently, two paradigms allow users to explore large music collections: search and recommender systems. Utilizing naive search approaches based on simple attribute matching requires the collection data to be fully annotated with metadata. When relying on keyword search facilities, the user is required to have some idea of his/her current preferences and has to be able to formulate a query that actually describes these preferences well. More advanced search facilities are based on content similarities of items (aka "find similar artists or songs") and are rarely personalized. Especially data sparsity and the lacking ability for comparing heterogeneous items (tracks, artists, albums, etc.) makes it hard for such systems to succeed. In contrast, recommender systems propose items that might be suitable for the user (based on some collaborative filtering approach or more complex models. While recommender systems do not require the user to be able to formulate his/her current preferences, the user also is not able to directly influence recommendations by stating e.g., a starting point for his/her explorative search for music matching his/her current preferences (except for feedback mechanisms like relevance feedback and explicit ratings that influence the user model in the long term).

Only very few approaches like the one proposed by Chen et al. [1] allow the user to specify his/her current needs and preferences in an abstract manner, where the returned results are jointly based on the query (the user's current information need) and the user's personal music preferences. However, there is still a substantial lack of user interfaces that provide dynamic, exploration-driven visualization strategies for large collections of music.

Therefore, we propose the geMsearch system to bridge this gap in explorative music search. In particular, we propose to use graph embedding techniques for computing latent representations of items contained in the graph, such as tracks, users, artists, genres or acoustic features of tracks. Using such graph embedding techniques [14], a low-dimensional latent vector representation is learned for every node. These firstly allow to create advanced search facilities as search queries can be encoded in the same vector space. As a result, not only exact results can be retrieved, but also similar items and hence,

[1] http://press.spotify.com/us/about

MILC '18, March 11, 2018, Tokyo, Japan

exploiting previously unknown similarities between heterogeneous items that can be utilized to retrieve diverse search results. Secondly, the obtained vector representations can be exploited for advanced visualization paradigms enabling explorative music search.

This work presents a preliminary study and visualization prototype based on latent representations obtained by graph embedding techniques. In contrast to traditional list-based aggregations of search results that provide a one-dimensional view of the retrieved items, we exploit the low-dimensional vector representation to generate 3D representations of the suggested items, allowing users to visually explore the music collection in a 3D-space. The user is able to specify a starting point for his/her exploration of the musical 3D-space by browsing through this space, the query is implicitly refined and the user is provided with further suitable tracks and artists.

The remainder of this paper is structured as follows. In Section 2, we describe related work. Section 3 proposes a visualization for explorative music search based on graph embeddings and presents the proposed prototype. Section 4 sums up key aspects and details future work.

RELATED WORK

For the task of building visualizations for music exploration, there are a number of relevant approaches, mostly based on proximity-preserving dimension reduction techniques.

The Islands of Music interface [10] incorporates rythm descriptors and employs self-organizing maps for visualizing music collections based on the metaphor of geographic maps in two-dimensional space. One highly relevant extension of these maps is a browsable 3D landscape by Knees et al. [6], where tracks are clustered based on content features. Hamasaki and Goto [4] propose Songrium, a collection of visualization and exploration approaches. These include the "Music Star Map", a visualization of songs in a graph, where placement of songs is based on audio similarity. Also, Lamere et al. [7] presented a 3D interface (Search Inside the Music) based on Multidimensional Scaling (MDS) techniques to visualize similarities between tracks, where each item is represented as a single colored item in the 3D space. Similarly, the Music Box visualization approach relies on Principal Component Analyses to visualize tracks, where song similarity is used to distribute tracks on a plane. Stober et al. [13] also rely on MDS, however, utilize bisociative lens distortions to support serendipitous music discovery in the MusicGalaxy UI. The visualization proposed in this work differs from these approaches in the fact that we base the visualization on latent representations of items within a heterogeneous graph that includes tracks, artists, albums, genres, etc. Due to the applied graph embedding techniques, proximities within the graph visualization are not restricted to similarities between items of the same type (e.g., tracks) or similarities based on a single set of features (e.g., audio features), but rather capture the similarity of items of any type in the latent feature space.

Recently, graph embedding techniques have also been introduced to the field of music information retrieval. Chen et al. [1] utilize graph embeddings for realizing a query-based music recommender approach that is similar to the approach presented in this paper. A similar approach has also been utilized for playlist recommendation [2] or text-based music retrieval based on playlists [3]. However, these approaches do not provide a user interface for the exploration of new music.

GEMSEARCH: EMBEDDING-BASED VISUALIZATION

In the following section, we present the geMsearch system, a first prototype for personalized explorative music search based on latent representations of nodes of the musical ecosystem[2]. *geMsearch* stands for graph embedding based music search and consists of two main components, which we will detail in this section: the graph embedding and retrieval engine that computes latent representations of items and query results, and the client providing a search and visualization interface.

Graph Embedding and Retrieval Engine

For the creation of the graph underlying our approach, we rely on the Spotify playlist dataset by Pichl et al. [12], containing 852,293 tracks crawled from public Spotify playlists. To enrich the available item descriptors for improved query performance, we also add Last.fm tags[3] for the contained tracks. The resulting dataset is represented as a graph containing undirected edges between the following item types: user–track, track–tag, track–album, album–artist and artist–genre. For the computation of latent representations of nodes via graph embedding, we rely on the popular Deepwalk algorithm [11], where we learn representations for all nodes in a 128 dimensional vector space. The resulting latent representations provides means for flexibly computing similarities between heterogeneous items such as tracks, users or artists.

geMsearch allows users to interactively explore the music space to find new music. Therefore, a starting position for browsing through the items has to be determined by eliciting the user's current musical preferences. As can be seen in the top left corner of Figure 1, a text input field (with autocompletion support) allows to select multiple items from the dataset to construct a query that reflects the user's current preferences. Here, the search query for artist "Jimi Hendrix" may return similar and suitable artists, tracks or tags. In addition, the search result can further be refined by adding further search terms. In Figure 1, the tag "guitar" is entered and combined with the first term. To create a search vector which is evaluated to retrieve nearest neighbors as search results, the mean item representation of these query terms is computed. The scaled user's latent representation is finally added to this vector and hence, long-term preferences partly influences the outcome. The resulting vector is then used to retrieve the most similar items from the graph as search results.

Visualization

The most common visualization for both recommendation and search results is to display a list of items ordered by the predicted relevance of the individual items for the user. This limits users to only observing the sequential order of items and

[2]The prototype can be accessed at http://dbis-graphembeddings.uibk.ac.at

[3]https://www.last.fm/api/show/track.getTags

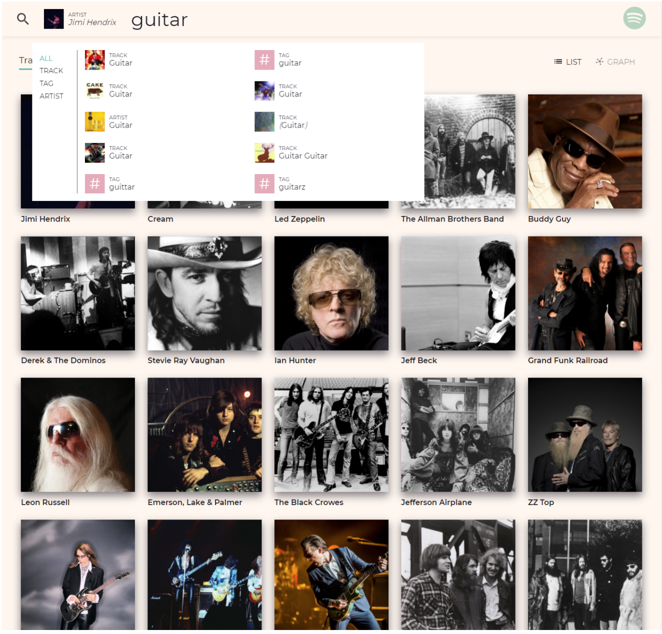

Figure 1. geMsearch query bar with autocomplete and list results.

hence, a one-dimensional view agnostic to distances between consecutive items. With a latent feature space underlying the system (obtained through, e.g., graph embedding techniques), similarities between arbitrary items can be expressed which permits developing more advanced interfaces. Through recent advances in browser technology, like the availability of native WebGl, just-in-time visualizations of 3D scenes can be created directly on websites without complex precomputations or add-ons. Using dimension reduction methods, the computed high-dimensional latent representations can be reduced to three dimensions, allowing to directly visualize items while preserving proximity. Here, we utilize principal component analysis to reduce the 128 dimensional representation of items to a three dimensional space. Instead of displaying a list of items, the recommended items can now be visualized in a 3D scene. Each track, artist or album can be positioned using its three-dimensional representation and can hence be displayed as an interactive 3D object. The positions and resulting distances reflect the relationships and proximities between items within the music collection. Beside the traditional list view for search results, the gemSearch client visualizes the surrounding items in a 3D WebGl scene as depicted in Figure 2. Using such an interface does not only allow to express distance between items, but, more importantly, it allows the user to explore and browse through the result space interactively. Mouse gestures allow for exploring the virtual space and while navigating, additional items are lazy-loaded into the scene.

The user may first use a keyword search to express his/her current preferences (cf. section on Graph Embedding and Retrieval Engine). Based on these criteria, the first search results are retrieved and displayed in a 3D space, where the user should feel like navigating through a virtual result space instead of jumping to unconnected items. In the underlying latent vector space, any of the proposed items can be used to further extend the query and hence, refine the search to match current preferences more precisely. Besides this active manipulation, the 3D scene provides an even more effective process of implicit refinement. The most relevant search results are positioned around the center of the screen. When exploring additional items further away, the user has to opt for a direction in which to continue exploring. After inspecting items at the new position, the navigation direction can be refined. If the user detects suitable items, the direction is correct; otherwise the user will navigate in a different direction. This choice of directions and moving within the virtual result space directly translates to (implicit) query refinement.

It is crucial to simplify the inspection of single items such that huge collections of music are explorable in reasonable time. We use album covers as textures for 3D objects describing track and album items and hence, also allow for visually inspecting node textures as this has shown to be an efficient means for judging the relevance of albums and tracks [9]. To provide detailed information about selected items (e.g., artists of a given track, genres, etc.), information from the underlying graph is retrieved and displayed. Also, we provide music previews for each track that allow users to inspect and immediately consume newly discovered tracks.

As similar items are located in close proximity to one another in the resulting space, distance-based clustering techniques can be applied to represented accumulations of items as annotated clusters. This allows users to decide whether a set of items might be of interest by looking at the characteristics of the cluster and not having to inspect the individual items contained in the cluster. However, zooming in into a cluster to inspect the individual contained items is still possible. Figure 2 shows how clusters of similar items are represented as single orange circles. On click, the contained items are shown while all other elements are faded with transparency to enhance the contrast. As items within a cluster are positioned nearby, the scene is zoomed in without scaling the circle sizes to avoid overlapping elements.

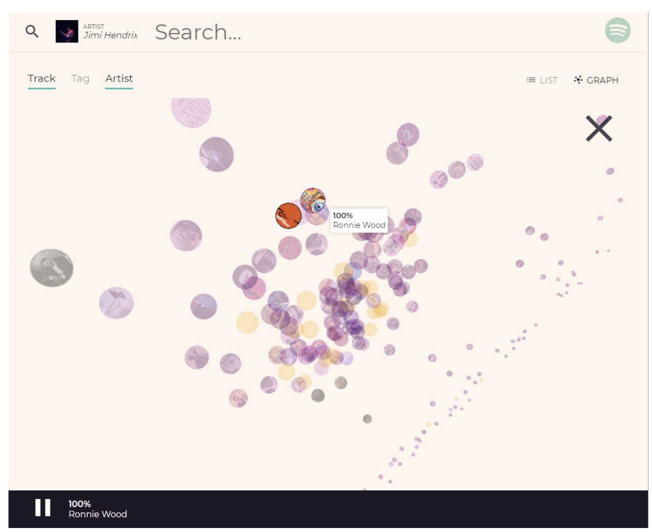

Figure 2. Web client 3D view and player bar.

To alleviate the cold start problem for user profiles, users can connect with their Spotify account. The official Spotify API supports the OAuth protocol with different scopes, allowing access to, e.g., personal playlists, playing history or saved tracks. To create a personal preference profile, *geMsearch* retrieves the user's saved tracks as we argue that saved tracks may serve as a strong indicator for preference. After a user has connected with his/her account, the music library is loaded and compared with the current contents of the underlying graph. For tracks, artists, etc. that are not yet contained in the underlying graph, we gather the missing metadata from Spotify and user-curated tags describing these items from Last.fm. After the data is collected, the graph is extended with this new information. In a next step, latent representations have to be computed in case of new items or updated in case of items that are affected by the newly added information. Deepwalk uses short random walks to model the graph structure with an uniform distribution over nodes. Therefore, neither the complete graph structure, nor all nodes have to be known to the algorithm initially. This implies that additional nodes and edges can be added on-the-fly to continue learning and extending existing embeddings without the need to relearn the complete model from scratch when adding new users or items.

CONCLUSION AND FUTURE WORK

In this work, we presented *geMsearch*, a preliminary prototype for personalized exploration and search of music collections. We exploit graph embedding techniques to compute a low-dimensional vector space representation of the music collection and the contained items. This allows for query-based, personalized exploration of music collections. Particularly, our approach provides users with a 3D representation to yield a visual exploration of new music; allowing the user to browse through search results and the full collection, where the distance of items (tracks, artists, genres) in the displayed graph corresponds to item similarity. Please note that the browsing through the 3D-space is not restrained to search results, the user's query is a mere definition of a starting point for browsing for the full collection graph and hence, query refinement.

We believe that the proposed method is not necessarily limited to music and may also be used in different domains, where data can be represented as graph and metadata for single items is sparse.

As for future work, we aim to further extend the prototype by improving the visualization performance and updating user preferences on-the-fly. For computing the user profiles, we aim to look into incorporating listening histories and create more comprehensive user profiles. Also, we aim to lay a particular emphasis on interaction aspects in the prototype by, e.g., allowing the user to up- or downvote certain tracks explicitly. We further aim to perform a user-centric evaluation of the system.

REFERENCES

1. Chih-Ming Chen, Ming-Feng Tsai, Yu-Ching Lin, and Yi-Hsuan Yang. 2016a. Query-based Music Recommendations via Preference Embedding. In *Proc. of the 10th ACM Conf. on Recommender Systems (RecSys '16)*. 79–82.
2. Chih-Ming Chen, Chun-Yao Yang, Chih-Chun Hsia, Yian Chen, and Ming-Feng Tsai. 2016b. Music Playlist Recommendation via Preference Embedding. In *Proc. of the Poster Track of the 10th ACM Conf. on Recommender Systems*.
3. Chia-Hao Chung, Yian Chen, and Homer Chen. 2017. Exploiting Playlists for Representation of Songs and Words for Text-Based Music Retrieval. In *Proc. of the 18th Intl. Society for Music Information Retrieval Conf.*
4. Masahiro Hamasaki and Masataka Goto. Songrium: A Music Browsing Assistance Service Based on Visualization of Massive Open Collaboration Within Music Content Creation Community. In *Proc. of the 9th Intl. Symposium on Open Collaboration*.
5. Mohsen Kamalzadeh, Dominikus Baur, and Torsten Möller. A survey on music listening and management behaviours. In *Proc. of the 12th Intl. Society for Music Information Retrieval Conf.*
6. Peter Knees, Markus Schedl, Tim Pohle, and Gerhard Widmer. 2006. An innovative three-dimensional user interface for exploring music collections enriched. In *Proc. of the 14th ACM Intl. Conf. on Multimedia*. ACM, 17–24.
7. Paul Lamere and Douglas Eck. 2007. Using 3D Visualizations to Explore and Discover Music. In *Proc. of the 7th Intl. Society for Music Information Retrieval Conf.* 173–174.
8. Jin Ha Lee, Yea-Seul Kim, and Chris Hubbles. 2016. A Look at the Cloud from Both Sides Now: An Analysis of Cloud Music Service Usage. In *Proc. of the 17th Intl. Society for Music Information Retrieval Conf.* 299–305.
9. Janis Libeks and Douglas Turnbull. 2011. You can judge an artist by an album cover: Using images for music annotation. *IEEE MultiMedia* 18, 4 (2011), 30–37.
10. Elias Pampalk. 2001. *Islands of music: Analysis, organization, and visualization of music archives*. Master's thesis. Technical University Vienna.
11. Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proc. of the 20th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*. ACM, 701–710.
12. Martin Pichl, Eva Zangerle, and Günther Specht. 2017. Improving Context-Aware Music Recommender Systems: Beyond the Pre-filtering Approach. In *Proc. of the 7th Conf. on Multimedia Retrieval (ICMR)*. ACM, 201–208.
13. Sebastian Stober, Stefan Haun, and Andreas Nürnberger. 2012. *Bisociative Music Discovery and Recommendation*. Springer Berlin Heidelberg, 472–483.
14. Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. 2007. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 1 (2007), 40–51.