Social Audio Features for Advanced Music Retrieval Interfaces

Michael Kuhn
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
kuhnmi@tik.ee.ethz.ch

Roger Wattenhofer
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
wattenhofer@tik.ee.ethz.ch

Samuel Welten
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
swelten@tik.ee.ethz.ch

ABSTRACT

The size of personal music collections has constantly increased over the past years. As a result, the traditional metadata based lists to browse these collections have reached their limits. Interfaces that are based on music similarity offer an alternative and thus are increasingly gaining attention. Music similarity is typically either derived from audio-features (objective approach) or from user driven information sources, such as collaborative filtering or social tags (subjective approach). Studies show that the latter techniques outperform audio-based approaches when it comes to describe the perceived music similarity. However, subjective approaches typically only define pairwise relations as opposed to the global notion of similarity given by audio-feature spaces. Many of the proposed interfaces for similarity based music access inherently depend on this global notion and are thus not applicable to user driven music similarity measures. The first contribution of this paper is a high dimensional music space that is based on user driven similarity measures. It combines the advantages of audio-feature spaces (global view) with the advantages of subjective sources that better reflect the users' perception. The proposed space compactly represents similarity and therefore is well suited for offline use, such as in mobile applications. To demonstrate the practical applicability, the second contribution is a comprehensive mobile music player that incorporates several smart interfaces to access the user's music collection. Based on this application, we finally present a large-scale user study that underlines the benefits of the introduced interfaces and shows their great user acceptance.

Categories and Subject Descriptors

H.5.1 [**Information Systems**]: Multimedia Information Systems

General Terms

Algorithms, Experimentation, Human Factors

MM'10, October 25–29, 2010, Firenze, Italy.

1. INTRODUCTION

The way we interact with music has drastically changed over the past years. Reasons are the online availability of music, compact media formats, and ever increasing storage capacities. These technical advances also facilitated a shift towards the mobile domain.

Unfortunately, the interfaces offered by (mobile) media players lag behind this trend of ever growing collections. Today, the organization of digital music libraries is mostly handled with metadata included in the audio files. Traditional list based search and browsing options render it hard to keep an overview over a large amount of tracks. As a result, users experience problems selecting the appropriate music for a given mood or situation. Research about the users' needs has shown that people are searching for music not only by means of bibliographic data (i.e. artist and title), but also in more descriptive ways, such as by specifying genre or mood information, or by naming artists that are similar to the desired music [5, 12, 23]. However, current music players do typically not support unspecific search queries like: "I want to listen to music similar to the current track, but not of the same artist", or "I would like to listen to something happy".

Search methods based on *music similarity* offer an alternative that allows users to abstract from manually assigned metadata. The similarity based organization has the advantage of providing easy navigation and retrieval of new items, even without knowing the songs by name. Online services like last.fm and iLike that enjoy great popularity show the potential of similarity based music retrieval. A considerable amount of research has been devoted to the development of interfaces to make similarity information intuitively available to the end user. Even though a variety of advanced interfaces for collection visualization and playlist generation have been proposed, they have thus far not reached the masses. A possible reason for the limited success is that many approaches are designed for the use with audio feature spaces. However, research indicates that the capability of audio analysis to represent perceived music similarity is limited. Methods based on implicit and explicit user feedback have been shown to better reflect the users' perception, which is confirmed by the success of the aforementioned online services.

There are two main reasons for the popularity of audio-features for interface design. First, their compact representation facilitates the use in mobile devices, without the need for permanent Internet connectivity. Usage data based similarity measures, on the other hand, typically require large

databases to answer queries. Second, many of the currently available interfaces rely on a global perspective on music similarity. Such a global view is, for example, extremely advantageous in the context of collection visualization. However, socially derived music similarity typically only provides a local view, i.e. pairwise similarities in a close neighborhood of a given query song. The lack of a global view greatly complicates the design of interfaces.

In this paper we propose the notion of *social audio features* to overcome these issues. Our social audio features combine the advantages of audio feature spaces and subjective music similarity measures. In particular, we construct a high-dimensional music space that well reflects the pairwise similarities gained from subjective measures. This space can be used by interfaces in the same way as traditional audio-features spaces. Our approach exploits two sources of information to derive music similarity: The users' listening behavior and social tagging. In particular, we describe a method that combines the two signals before they are fed into a statistical framework called Probabilistic Latent Semantic Analysis (PLSA). PLSA basically performs dimensionality reduction on co-occurrence data by identifying a (small) set of hidden variables that well explain the observed co-occurrence values.

The resulting social audio features cover more than 1M songs. On the artist level, we managed to even map roughly 1.4M artists. These numbers clearly exceed approaches that head into similar directions, and facilitate the use in end user applications. To demonstrate the practical usefulness, we have implemented a comprehensive mobile music application (*museek*) for the Android platform. *museek* applies our social audio features in different similarity based interfaces. Enhanced with album art, for example, the underlying space facilitates the visual browsing of a collection, and by considering skipping behavior, it allows to quickly identify undesired regions. Moreover, the application is aware of the social tags used to derive music similarity. In combination with the map, these tags enable a fine-grained selection of music matching a certain mood or style.

In an analysis of more than 100 usage data logs we show the necessity of such advanced music retrieval interfaces. In line with studies on music web forums, we find that, while traditional search options remain important, less specific and more explorative ways to access music are frequently used and highly accepted by the users.

2. RELATED WORK

When designing music retrieval systems, it is essential to know about the needs of the end users: How do people search for music? What tools could assist them to find what they are looking for? To get a better understanding of the users' needs, Bainbridge et al. [5] have analyzed music queries in the *Google Answers* service. Not surprisingly, artist and song title are most often used to search or ask for music. However, the study also shows that roughly a third of the queries included a description of the genre or style, and that sometimes references to known similar musicians were given. Similar results were reported by Downie and Cunningham [12] in an study on newsgroup messages. These results are in line with the findings of Bentley et al. [6], that suggest that music retrieval systems could profit from support for serendipitous browsing capabilities. Lee et al. [23] used questionnaires to investigate how people are searching for music. Their study even more clearly emphasizes the importance of non-specific search options, such as genre, mood, or gender. While genre information is doubtlessly important, there is only little agreement on genre assignments and taxonomies [3, 21], which limits its usefulness. An alternative to genres is given by social tags that provide several advantages [21]: They overcome synonymy issues (e.g. E-Jazz vs. Electronic Jazz), allow for a fuzzy assignment (a song might, e.g., be tagged with Nu-Jazz and E-Jazz), and can also grasp mood and other non-genre related information.

To facilitate non-specific search and browsing it is helpful to have an understanding of music similarity. The various research approaches that address this issue are roughly classified in acoustic and subjective approaches by Berenzweig et al. [7]. Subjective measures encompass any kind of techniques that involve human interaction such as the analysis of expert assigned metadata, collaborative filtering based techniques, questionnaires, and so on. Acoustic approaches, on the other hand, solely rely on audio-signal analysis and do not involve any human judgment, and are thus an objective measure. Examples are the works of Logan and Salomon [28], Aucouturier and Pachet [1], Foote [14], Pampalk et al. [32], Tzanetakis [43], and Tsunoo et al. [41]. A good overview and discussion of audio based techniques is given in [9].

Audio based measures have the advantage that they are not influenced by any subjective bias. Moreover, the extracted features typically define some sort of a space that provides a powerful basis to construct novel interfaces to access music. Audio-analysis, however, also exhibits some major disadvantages. Although objectivity might be advantageous in some scenarios, the lack of subjective information proves to be a problem in most real-world settings. After all, music is typically targeted at people, and the perception of music is inherently subjective. Thus, many approaches try to find a mapping between audio features and some widely used genre taxonomies (see e.g. [43] and [41]). However, Aucouturier and Pachet [4] conjecture that the currently used techniques and their variants have reached a plateau with respect to accuracy that can not easily be overcome. Moreover, they show that errors produced by state-of-the-art methods are often severe, i.e. misclassified songs can be completely different from their neighbors in terms of perceived similarity. The conclusions of Casey et al. [9] go into a similar direction.

Subjective methods encompass a wide variety of techniques. Sources of information include (expert assigned) metadata (e.g. [35]), (web-)text documents describing music (e.g. [45]), questionnaires (e.g. [7]), games (e.g. [13]), and usage data (e.g. [16]). Questionnaires are mainly used as a ground truth to compare other methods, such as in [7]. Sometimes, different techniques can also be combined. There are, for example, games that aim at collecting metadata (e.g. [22] and [29]) that can in turn be used to define music similarity. Social tags are used as metadata in the approach presented by Levy and Sandler [25]. The authors stress the advantages of social tags over web-mined information with respect to noise and scalability. The results of Turnbull [42], however, show that social tags suffer from a so called popularity bias, i.e. they offer good quality for famous songs, but are of limited use to describe less known items.

In an attempt to define a notion of ground truth, Berenzweig et al. [7] compare different methods to derive music

similarity. The results indicate that co-occurrence information in conjunction with item-to-item collaborative filtering [26] provides the best results among the subjective techniques. In the experiments of Slaney and White [39], collaborative filtering also clearly outperforms an audio-based alternative.

These studies indicate that collaborative filtering is superior to other approaches when it comes to define (perceived) music similarity. However, the resulting information is not as versatile as the feature-spaces produced by audio-analysis. In particular, item-to-item collaborative filtering only produces a weighted list of neighbors that co-occur at least once with the item in question. Hence, the method does not define a global space. Such a space, however, is required by many advanced interfaces. This is, presumably, one of the reasons, why most interfaces that have recently been proposed are based on audio-feature spaces.

With the work we present in this paper, we bridge this gap. In particular, we propose a method to create a music similarity space based on social tags and ideas from collaborative filtering. The single information sources have been used in comparable contexts before, such as in [25] (social tags) and [10, 16] (collaborative filtering). The existing collaborative filtering based approaches, however, miss the advantages of tags that have an intuitive meaning and can thus efficiently guide the user through the space. Moreover, the approach of Gleich et al. [10] is targeted merely at visualization and limited to artist similarities.

The approach of Levy and Sandler [25] solely relies on social tags. Our approach adds to this work by including ideas from [16] to overcome the popularity bias problem reported by Turnbull et al. [42]. Our experiments show that the combination of the two information sources does not only allow to cover enough songs and artists to make our space practically applicable, but that it also significantly improves the accuracy of the music similarity information. Moreover, the resulting space keeps the advantages of the tags and their intuitively understandable meanings. The result is a music similarity space that can be used in a similar way as audio-feature spaces.

Most of the existing work in the context of user interfaces tries to visualize collections based on previously extracted audio features. Often, self-organizing maps (SOMs) are used for this purpose, such as in [30, 20, 24, 15]. These approaches typically map the audio space into some low (2 or 3) dimensional representation, which can then be explored using traditional navigation methods. Other approaches that go into a similar direction include the work of Donaldson and Knopke [11], and Sony Ericsson's commercial SensMe interface that displays songs along two axes (mood and tempo). How this information is extracted remains the company's secret. Moreover, several more abstract visual interfaces have been proposed, such as the artist map of van Gulik et al. [44], and the circular layouts introduced by MusicRainbow [33] and AudioRadar [18]. These interfaces all rely on audio features as an underlying similarity measure, sometimes augmented with metadata. The interface proposed by Torrens et al. [40] does not rely on any music similarity measure. Rather, it directly visualizes metadata, such as genre or year of release. Apple's Cover Flow interface, finally, merely replaces the traditional textual album list by nicely presented album covers. The enormous popularity indicates that album covers are a useful visual hint to retrieve music.

Often, visual interfaces are combined with intelligent playlist generation. Examples are the commercial SensMe interface, the approach of van Gulik et al. [38], and the PocketSOMPlayer [15] that allows to create playlists by drawing trajectories through a SOM-based map.

Non-visual playlist generation methods have been proposed by Pampalk et al. [34] and Bossard et al. [8]. These approaches try to find music that matches the user's taste by considering feedback such as skipping behavior.

A purely textual interface to generate intelligent playlists is Apple's iTunes Genius feature. It basically selects songs similar to the preceding songs and thus follows a similar principle as approaches presented in [27], [36], [2], and [37]. In *museek* we have implemented various play modes and a visual browsing interface that incorporate several of the outlined ideas.

3. CREATING A SOCIAL AUDIO SPACE

Probabilistic Latent Semantic Analysis (PLSA) is a statistical framework to analyze co-occurrence data. The method was proposed by Hofmann [19], and originally designed for automated document indexing. Similarly as in the widely known Latent Semantic Indexing (LSI) approach, the idea is to discover relationships between words and documents in a document collection. While LSI relies on techniques from linear algebra for this purpose, PLSA makes use of a probabilistic model. In particular, it introduces latent variables (also called latent classes) that interrelate words and documents. In a comparative study, Levy and Sandler [25] have found that PLSA provides better results than LSI in a context very similar to ours.

The goal of PLSA is to find probabilistic assignments between documents and latent classes, and between latent classes and words such that the observed occurrences of words in documents is best possible approximated by the probabilistic model. Hofmann shows that the corresponding assignments of documents to latent classes can be interpreted as a vector space. Thus, every document can be seen as a point in this space, and, as a result of the analysis, similar documents reside at similar locations in this space (we use the L_2-norm to measure distance).

Analogously, a latent semantic music space can be constructed by considering songs as documents, and the social tags assigned to these songs as the words within the "documents". The result is a space of music in which similar songs are supposed to cluster.

3.1 PLSA

To describe the PLSA method we will make use of the notation used for document indexing, i.e. we will talk about *documents* d, *words* w, and *latent classes* z. We will thereby assume that there are N documents, M words, and K latent classes.

In PLSA, documents are related to words via latent classes. Thereby, a generative model is assumed, in which a document is created by producing its words as follows: Each word is generated by first choosing a latent class, and then, dependent on the latent class, choosing a word. In particular, for each word, first, a latent class is chosen with a certain probability $P(z_k|d_i)$. Dependent on this latent class, then, a word is chosen according to the probability $P(w_j|z_k)$. In

this model, the probability that a document d_i creates a certain word w_j is given by:

$$P(w_j|d_i) = \sum_{k=1}^{K} P(w_j|z_k) \cdot P(z_k|d_i)$$

PLSA tries to find the assignment of the corresponding probabilities that best approximates the effectively observed document-word co-occurrences. The optimization of the model parameters (i.e. the probabilities) is done using the well known Expectation Maximization (EM) technique that works by alternately applying an expectation (E) and a maximization (M) step. The iterations are aborted as soon as some convergence criterion is met (between 20 and 50 iterations have shown to be sufficient in practice). In the context of PLSA, the goal is to maximize the log-likelihood L of the observed data, which is given by

$$L = \sum_i \sum_j n(d_i, w_j) \cdot \log P(d_i, w_j),$$

where $n(d, w)$ denotes the number of times word w occurred in document d. The corresponding expectation and maximization steps are given below:

- Expectation step:

$$P(z_k|d_i, w_j) = \frac{P(w_j|z_k) \cdot P(z_k|d_i)}{\sum_{l=1}^{K} P(w_j|z_l) \cdot P(z_l|d_i)}$$

- Maximization step:

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j) \cdot P(z_k|d_i, w_j)}{\sum_{m=1}^{M} \sum_{i=1}^{N} n(d_i, w_j) \cdot P(z_k|d_i, w_m)}$$

$$P(z_k|d_i) = \frac{\sum_{j=1}^{N} n(d_i, w_j) \cdot P(z_k|d_i, w_j)}{n(d_i)},$$

where $n(d_i)$ denotes the total number of words in document d_i.

3.2 Applying PLSA to Music

To create our social audio features, we have applied the PLSA method to data gathered from *last.fm*. Thereby, we consider two sources of information: (1) Social tags, such as they were assigned by users to songs and (2) the listening behavior of the users. The listening behavior is extracted from lists that, for each user, contain the 50 most listened songs (short *top-50 lists*). The following work is based on crawled data from about 2.4M users, containing approximately 10M songs, 1.4M artists, and 1M tags.

Observe that there are different ways how this information could be used in conjunction with PLSA, which basically only requires some sort of co-occurrence data:

- Using user-song co-occurrences, similarly as this was done in the graph embedding based map proposed by Goussevskaia et al. [16].
- Using the co-occurrence of songs and social tags (as described in [25]).

Using the song-tag co-occurrences is an intuitive approach and has proven to work well. We improve upon this basic approach by smartly re-assigning tags prior to applying PLSA. Thereby, we implicitly take advantage of the user-song co-occurrences, and thus effectively combine the two information sources. In the evaluation part, we will show that this combination leads to a significant performance gain as compared to approaches that only consider a single information source.

In the following, songs are considered as documents and (re-assigned) tags are considered as words in the context of PLSA. Thus, for the remainder of this section, we will change our notation as follows:

- document $\rightarrow$ song: $d \rightarrow s$
- word $\rightarrow$ tag: $w \rightarrow t$

The tags assigned by *last.fm* users exhibit some peculiarities. In particular, there are lots of *personal tags* that relate to the user who assigned them rather than to the music. Examples are "heard on pandora", "favorite artist", and "awesome". Moreover, several spellings are used to denote the same thing, such as "hip hop", "hip-hop", and "hiphop". To reduce noise, we only considered the approximately 1K most occurring tags and manually cleaned them by removing *personal tags* and by normalizing synonyms. The reduction to the most occurring tags on the one hand allows for manual cleaning, and on the other hand considerably reduces the computational complexity without significantly affecting the accuracy.

A bigger issue with social tags is that they suffer from a popularity bias as shown by Turnbull et al. [42]. That is, typically only the most famous songs are accurately tagged, whereas unpopular songs often contain no, or inappropriate tags. In fact only about 20% of the songs in our last.fm subset were tagged by the users. Directly applying PLSA to the song-tag co-occurrence data, such as done in [25], would thus exclude the remaining 80% of the songs, which is not acceptable for the use in real-world applications.

To overcome this problem, we make use of the information contained in the users' top-50 lists. Similarly as in item-to-item collaborative filtering [26] we assume that songs that often occur together in such top-lists are related to each other to a certain degree. This assumption facilitates the extrapolation of the tagging information to previously untagged songs. In Section 3.3 we will show that this does not only solve the data sparsity problem, but also improves the accuracy of the resulting space.

In particular, we automatically assign tags to a given song s using the following procedure: Loop through the top-50 lists of all users. For each top-list song s appears in, iterate through all the neighbors (i.e. through all the other songs in the corresponding top-list). For each neighbor, iterate through all its tags (i.e. all the tags that last.fm users have assigned to this particular song). For each of these tags, increase the song-tag co-occurrences of song s. Finally, assign the T tags with highest occurrence to song s for the use in the PLSA optimization (for some threshold T, 50 in our case). Moreover, these tags are weighted proportionally to their occurrence numbers (and such that the weights sum up to 1). This automated tag assignment process is illustrated in Figure 1 for a simplified example with only two top-lists (and $T = 3$).

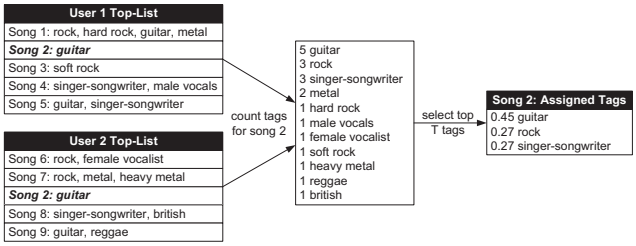


Figure 1: Assigning weighted tags to a song (simplified example with only 2 top-lists).

We have tagged the (approximately) 1.1M most occurring songs using the described technique. The obtained song-tag co-occurrence data has then been fed into the PLSA framework. After applying PLSA, the conditional probabilities $P(z_k|s_i)$ are well defined for all classes z_k and all songs s_i. These probabilities can be seen as coordinates, assigning each document a point in the so called *probabilistic latent semantic space* [19]. Since the probabilities corresponding to a song s_i sum up to 1 (i.e. $\sum_k P(z_k|s_i) = 1$), the songs in fact lie on a $K-1$ dimensional hyperplane. The coordinates of the songs in this space form our *social audio features*.

The resulting space covers roughly 1.1M songs corresponding to more than 120K artists. However, our *last.fm* database contains information to more than 1.4M artists. To make this information available we have calculated *artist coordinates* for these artists as follows: (1) For all the artists that are available in the latent space, we define the *artist coordinate* as the center of mass of their songs with known coordinates. (2) For the remaining artists we have queried their closest neighbors from *last.fm*. We then place the artist at the center of mass of all the neighbors with known coordinates (as calculated before). As a result we could define social audio features for more than 1.4M artists, which is enough to facilitate the use in end-user applications.

An important property of our space is the direct relationship to the tags that were used in its construction process. In particular, the PLSA-model inherently defines the probability of a song generating a given tag as:

$$P(t_j|s_i) = \sum_{k=1}^{K} P(z_k|s_i) \cdot P(t_j|z_k)$$

We will make use of this relationship in the mobile application presented in Section 5.

3.3 Evaluation

In this section we describe different tests that allow to compare the qualities of different music similarity spaces. In particular, we will compare the spaces resulting from the following approaches:

- *Song-tag approach*: PLSA is applied to plain song-tag co-occurrences (as given from last.fm).
- *Song-user approach*: PLSA is applied to song-user co-occurrences.
- *Combined approach*: PLSA is applied to the co-occurrences of songs and re-assigned tags, as described before.

Since the song-tag approach suffers from the mentioned popularity bias problem, the comparison is done on a reduced data set. Similarly as in [25] we only consider songs that contain at least 30 tags. To ensure fairness with respect to the *song-user approach*, we have also eliminated songs that appear in less than 30 top-50 lists. The resulting *reduced dataset* contains roughly 80K songs. To construct the space with the *combined approach* we have applied PLSA to both, the reduced as well as the full dataset. To keep the numbers comparable, the same 80K songs were used for both datasets during evaluation.

Our tests are based on three different criteria to assess the quality of a given music space:

- *Consistency of social tags*: In a space that well reflects perceived music similarity, the social tags of songs in a close neighborhood should be similar.
- *Comparison to collaborative filtering*: Collecting a sufficient amount of human judgments to get a *ground truth* with respect to perceived similarity is an extremely expensive task. Moreover, there do not seem to be any publicly available datasets that can be used for this purpose. Thus, we rely on item-to-item collaborative filtering as a "ground truth" to which we can compare our results. Berenzweig et. al [7] have compared a variety of music similarity measures and found that item-to-item collaborative filtering performs best among the investigated approaches.
- *Artist clustering*: Songs of the same artist are often similar. Thus, songs of the same artists are supposed to (somewhat) cluster in a space that well reflects music similarity.

3.3.1 Tag Consistency

To evaluate tag consistency, we try to estimate the (uncleaned) tags of a given song by considering the (uncleaned) tags of songs residing in its neighborhood. For this purpose, we closely follow the concept of a k-nearest-neighbor (KNN) classifier. To estimate the tags of a song s, we consider the tags assigned to the k closest songs in the music space (for $k = 20$). For each tag we count the total number of occurrences. The 10 most occurring tags are then compared to the 10 tags that were most often assigned to s by *last.fm* users. The percentage of correctly estimated tags is a good measure to compare different music spaces to each other. We have not only used it to compare the *combined approach* to the other two PLSA variants, but also to decide on an appropriate number of latent classes (i.e. the dimensionality of the resulting latent semantic space). Figure 2 plots the number of latent classes versus the percentage of correctly estimated tags. As expected, the number of latent classes significantly influences the accuracy of the resulting music space. An important observation is that the curve levels off. That is, increasing the number of dimensions beyond about 30 does not lead so a significant increase in the precision of the estimated tags. Thus, we have fixed the number of dimensions to 32.

We have measured the tag consistency on the three PLSA variants (song-tag, song-user, and combined) on the reduced dataset, and, in addition, on the combined variant on the full dataset. The results are summarized in Figure 3. On the one hand, we can see that the combined variant significantly

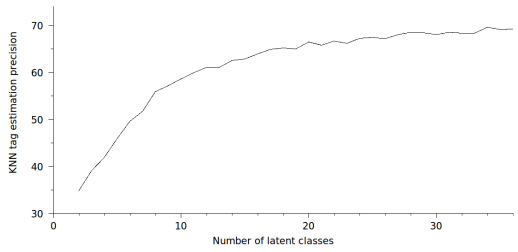


Figure 2: The number of latent classes versus the percentage of correct tag estimations. The plot shows that increasing the number of latent classes to more than about 30 does not lead to a relevant quality improvement.[2]

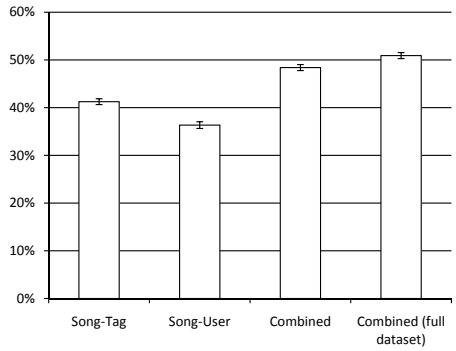


Figure 3: Tag consistency: Combining the two information sources (social tags and listening behavior) improves the KNN tag estimation.

improves the tag estimations as compared to the simple approaches. On the other hand, the figure shows that on the full dataset the performance even increases. Over half of the most relevant user-assigned tags could be correctly estimated by looking at the neighborhood of a song – a remarkable number when considering the synonym issues and the many personal tags present in the last.fm dataset.

3.3.2 Comparison to Collaborative Filtering

The *top-50 lists* available from our last.fm dataset can be used to calculate a ranked neighborhood of a song using item-to-item collaborative filtering (our "ground truth"). Collaborative filtering is designed to identify the most similar items and is known to perform well in this respect (recall Section 2). However, for most distant items, it does not provide any information due to the lack of co-occurrence information. In our experiments, we thus compare the k closest neighbors in our map with the k most similar items identified by item-to-item collaborative filtering, applied to our *top-50 lists*.

The results of applying this measure to the different space construction variants are shown in Figure 4. Again, we can see how the combination of the two signals significantly improves the quality. The approximately 25% matches in the 10 closest neighbors should be contrasted to the 80K songs

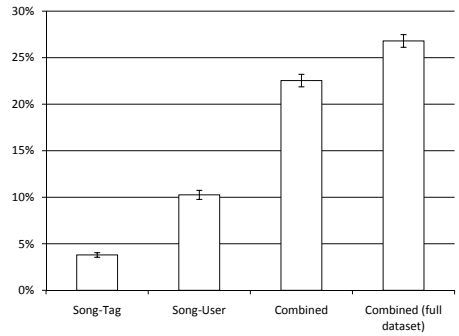


Figure 4: Comparision to collaborative filtering: The combined approach clearly outperforms the alternative approaches. More than 25% agreement is a remarkable number, considering that only 10 songs were compared from a universe of 80K songs.

these neighbors could be chosen from. The task is comparable to the search for a needle in a haystack – thus, more than 25% identical output are a remarkable result.

3.3.3 Artist Clustering

We measure the level of artist clustering using the mean average precision (mAP) on artist labels. Average precision (AP) is a standard performance metric known from information retrieval. It is used to measure the quality of a ranked sequence of items, such as given when ordering songs according to their distance from a given query song s. Thereby, relevant items (songs featuring the same artist as the query song, in our case) that appear early in the list are rewarded more than those that appear towards the end. More formally, AP is defined as

$$AP = \frac{\sum_{r=1}^{N} P(r) \cdot \mathrm{rel}(r)}{R},$$

where $P(r)$ denotes the precision at rank r, $\mathrm{rel}(r)$ is 1 if the item at rank r is relevant (and 0 otherwise), R is the total number of relevant items, and N is the total number of retrieved items (i.e. all the songs, in our case).

Higher AP (and thus also mAP) values refer to better artist clustering. However, a better artist clustering does not necessarily imply a better quality of the underlying space, as there is no reason, why songs of other artists cannot be similar to the query song. *John Lennon* and *Beatles* might serve as an example. In the same way, a single artist can have songs of extremely different style (e.g. *Nothing else Matters* and *Master of Puppets* from *Metallica*). Thus the mAP performance metric is questionable in our context, in particular when comparing relatively high mAP values. As it has been used before to quantify the accuracy of music similarity (see, e.g. [25]), we will apply it as well, despite its questionable nature.

The corresponding results are summarized in Figure 5. In line with the previous results, the figure shows that the two information sources (listening behavior and tags) can profit from each other. And again, the result of the combined

[2]The absolute numbers cannot directly be compared to the other experiments, as this plot is based on a different dataset and different parameter settings.

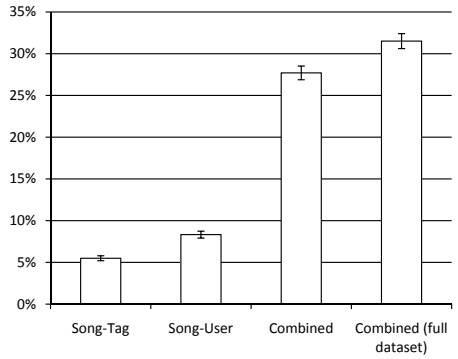


Figure 5: Artist clustering: Combining the two information sources significantly improves the mean average artist precision. Observe that this test is based on a total of 11K artist labels, which inherently leads to relatively low numbers.

approach even improves for the full dataset. When comparing these numbers to other approaches, it is important to consider the number of artists in the dataset. In [25], for example, the dataset contained 212 artists, as opposed to roughly 11K in our experiments (which inherently leads to lower mAP values).

4. THE SOCIAL AUDIO FEATURES

In the previous section we have embedded social music similarity information into a Euclidean space. Each direction in this space reflects a concept as defined by the corresponding latent semantic classes. As a result, each point in the space well characterizes a style of music. The basic nature of our latent music space is thus identical to the nature of an audio feature space. However, whereas in an audio feature space, the directions reflect some acoustic properties (such as timbre, pitch, beat, etc.)[3], they reflect socially derived music concepts in our PLSA space. Thus, we refer to the corresponding features as *social audio features*, as opposed to the (normal) *audio features* that define a traditional audio feature space.

Working on a Euclidean space rather than on pairwise similarities exhibits several advantages. In particular, applications can benefit from the geometric properties and the resource friendly compact representation using coordinates.

Geometry.

Embedding music into a Euclidean space is not only an effective way to represent music similarity, but also allows to make use of the geometric properties for music players and retrieval interfaces. Interesting geometric elements are *trajectories*, *volumes*, and a *sense of direction*. The preferred music style of a user, for example, can often be compactly represented as a volume. Trajectories can be used to graphically generate playlists from one region to another (see e.g. [31, 17]). The sense of direction, finally, can be used to smartly extend existing playlists.

[3]These are just simple illustrative examples. Clearly, state-of-the-art audio-signal analysis methods are able to extract more complex acoustic properties.

Memory and Computation Time.

Calculating the similarity between two arbitrary pairs of items is typically a resource intensive task when working with social similarity information. In the case of item-to-item collaborative filtering, for example, the entire item-item co-occurrence matrix has to be stored. When considering any non-zero similarities as edges of a graph, the pairwise similarity of an arbitrary pair of nodes can be defined as the shortest path between them [16, 37]. However, this required shortest path calculation is expensive with respect to both, memory usage (the entire graph needs to be available) and computation time.[4] Using social audio features, on the other hand, provides basically the same similarity information at low cost: For each song, only a low-dimensional coordinate (32 in our case) needs to be stored (which is negligible when compared to the song's file-size). Moreover, many applications do not need to know about the similarity relationships among *all* the songs in the music universe. Rather, applications often operate on a small subset only, such as on a particular user's music collection. Using our social audio space, we only need to store the coordinates of the relevant songs on the target device, in order to make the corresponding similarity information available. Using pairwise similarities, by contrast, also requires to store all the intermediate songs such that distances can be defined transitively (i.e. by means of shortest paths in graphs). The obtained savings in terms of memory consumption are often immense, considering that collection sizes are typically in the order of thousands of songs, as opposed to hundreds of thousands of songs contained in our music space. The social audio features are thus a perfect match for use on personal collections in the mobile domain.

4.1 Key Facts

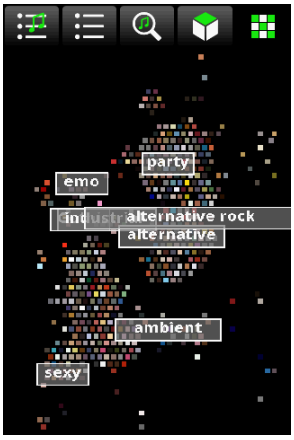
To conclude this section, we want to quickly review some major properties of the music similarity space that underlie our social audio features:

- *Similarity measure*: The underlying similarity measure is based on an analysis of *last.fm* data, combining the information from social tags and the user's listening behavior.
- *Orientation*: The axes in the described music similarity space are defined by latent classes which are directly related to the underlying tags. This information can be helpful to guide the user through the space.
- *Coverage*: The PLSA based space contains more that 1M tracks corresponding to more than 120K artists. To further increase the coverage, we have calculated the coordinates of an additional 1.3M artists, which leads to a significantly bigger coverage than any comparable approach. As a result, the coverage gets high enough to be usable in productive applications, such as the one shown in the following section.

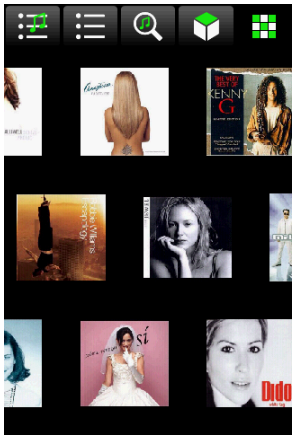
5. A COMPREHENSIVE MUSIC PLAYER

In an attempt to address the users' needs (as outlined in Section 2) and to demonstrate the usefulness of the social

[4]The concept of approximate distance oracles has been proposed to overcome these issues, however, in practice, the resource demands are still high.

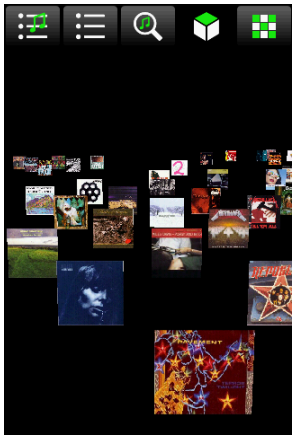


(a) The map in an overview zoom level.



(b) A close look at the similarity map.

Figure 6: The 2D music similarity map



(a) The music similarity map in 3D.

(b) An auto-generated tag-cloud for a user's collection.

Figure 7: The 3D map and the tag-cloud.

audio features, we have developed *museek*, a music player for Android smart phones, that provides similarity based functionality. Our player incorporates the following features to access and discover music: (1) Traditional alphabetic lists (song, album, artist, and genre) to browse for music, (2) a full text search option to search in the title and artist fields, (3) a tag-cloud to select music by social tags, (4) a music map to visualize a collection, (5) a play mode that plays songs similar to the previous one, and (6) a play mode that avoids inappropriate regions by considering skipping behavior.

Features (3) to (6) rely on the described social audio features. For visualization, our high dimensional space is transformed into a collection dependent 2D map on the user's device using Principal Component Analysis (PCA).

The traditional search options, i.e. features (1) and (2), do not need any further explanations. Rather, we quickly want to sketch the most important properties of the other interfaces. Our music map combines the strengths of Apple's Cover Flow and Sony Ericsson's SensMe interfaces. The popularity of Cover Flow shows that album art is a good visual hint to recognize music and also stresses the user's desire for visually attractive ways to browse a music library. SensMe, on the other hand, provides a neat way to explore a collection and to quickly create appealing playlists by selecting regions from a map. We have thus implemented an intuitively navigable map that uses album covers as visual aids.

At low zoom levels (see Figure 6(a)) the map resembles the point cloud of SensMe. Tags help the user to keep an overview. When zooming in, the points become recognizable as album images (see Figure 6(b)). From this view the user can either select an album to be played or discover other, similar albums by browsing through the covers. The navigation in the map occurs by moving the finger on the touchscreen. Zooming in and out can be done using traditional multi touch gestures (pinching and unpinching). Moreover, a touch gesture allows to select a region from the map to create a playlist similarly as in SensMe.

Besides the described 2D mode, the map also comes in a 3D flavor that focuses on offering an appealing browsing experience. The 3D view uses the same underlying 2D space but arranges the album covers in 3D (see Figure 7(a)).

We have also used the social audio features to offer two novel play modes, a *similar song mode* and a *smart shuffling mode*. The idea of the similar song mode is to extend an existing playlist with similar songs. This means a user can select a start song s/he likes and the application will automatically add songs to the playlist that are similar. This allows the generation of smooth playlists by choosing a single seed song. To avoid that the *similar song mode* plays songs from just one artist, the user can specify that an artist may not re-occur for a certain number of subsequent songs.

The smart shuffle mode selects songs from the entire collection and thereby intelligently avoids music styles of songs the user has previously skipped. Similar as in [8], the idea is to subdivide the map into good and bad regions. A region is marked good if the corresponding songs were listened to the end, and bad, if the songs were skipped.

Finally, we have seen that users often describe their needs in terms of genres or other descriptive information, such as mood. Thus we offer the possibility to choose songs by selecting a tag in a tag cloud (see Figure 7(b)). This tag cloud is individually generated for a user, displaying only tags that are relevant to the music collection on the device. As the tags in this cloud are freely generated by last.fm users, they do not only specify genres and sub genres but also moods and feelings. This facilitates a fine grained selection of the desired music.

The outlined functionality is integrated into our player in 5 tabs. The *Player Tab* contains the player controls, the playlist, as well as buttons to control the play mode (repeat, shuffling over playlist, shuffling over collection, similar songs, and smart shuffling). The *Lists Tab* allows to access traditional alphabetic lists (namely song, album, artist, and genre list), as well as a tag-cloud (recall Figure 7(b)) to select music. A full text search mode is provided by the *Search Tab*, and, finally, there are two screens for the 2D and the 3D map, respectively (recall Figures 6 and 7(a)).

6. USER STUDY

We have published our application on the Android Market (the App Store for Android). At the first startup, we ask the user for permission to log (anonymous) usage data. We removed overly short log files, as we were interested in the usage of regular users (as opposed to those that only had

	Explicit plays	Implicit plays
Meta data lists	16%	-
Similarity map	8%	-
Tag cloud	10%	-
Shuffle	-	2%
Smart shuffle	-	51%
Similar mode	-	13%
Sum	**34%**	**66%**

Table 1: A comparison between the origins of the played songs. The new player features are used often to generate playlists and select music.

a quick look at the application). The following statistics are based on the remaining 128 data logs, each of which documents the usage for a period of 5 days.

To get a rough impression about how the application is used, we measured the times the users spent in the different tabs. Not surprisingly, the Player Tab, mainly used to listen to music, is the most popular view – users spend about two thirds of the time in it. The remaining time can be seen as the time spent to search or browse for music and is distributed as follows: Lists Tab (including tag-cloud) 53%, Map Tabs 40%, and Search Tab 7%. The fact that when searching for music the users spent 40% of the time in the Map Tabs confirms the need for serendipitous browsing options and shows that this interface is well accepted.

Studies about user needs suggest that people would often select music based on descriptive information, such as genre or mood. We have found that 51% of our users have at least once selected music from our tag-cloud, and that 19% used this feature regularly. Interestingly, only about 40% of the selected tags correspond to some genre, the remaining 60% reflect some mood (e.g. "happy", "catchy") or subjective opinion (e.g. "beautiful", "amazing"). These numbers underline that genres alone are not descriptive enough to satisfy the users' needs.

The music player offers five different play modes: Repeat all songs of a playlist, shuffle over a playlist, shuffle over the whole collection, smart shuffling, and the similar song mode. The first two play modes define only the order in which a given set of songs is played. In contrast, the three other modes generate playlists by themselves, i.e. upon completion of a song they automatically select a new song to be played. Thus, we can distinguish between *explicit selection* of tracks (from the traditional lists, the tag-cloud, the search module, or the maps) and *implicitly generated suggestions* (from either collection shuffling, smart shuffling, or similar song mode). Table 1 shows how the listened songs are distributed among these different song selection methods.

Interestingly, only about a third of the music was explicitly selected by the user. The other two thirds were selected implicitly by the player, using one of the mentioned play modes. Moreover, we can see that the traditional search options account for less than half of the explicit selections, the remaining selections either occurred from the tag-cloud or the map. Considering the implicitly selected songs, we see that the similarity aware modes enjoy a great user acceptance. Surprisingly, the collection shuffling mode, which is prevalent in state-of-the-art players, is barely used.

Comparing these results to the studies of Bainbridge et al. [5], discussed in Section 2, it is no surprise that the traditional lists are the most popular means to explicitly select music. The usage numbers of the descriptive and visual browsing methods, however, are even higher than predicted. This might reflect the fact that people only become aware of certain retrieval techniques once they are provided with them, but then understand the advantages.

The plain numbers might let room for speculations. However, most of the results are clear enough to conclude that the novel features were well accepted by the users. Moreover, public feedback in the Android Market underlines the usefulness of similarity aware play modes. Considering smart shuffling, for example, people wrote: *"[...] Does a good job learning my tastes. [...]"* and *"Great app, learns what I like."*. Other comments confirm the acceptance of the similar song mode: *"[...] easy browse and make playlists. Auto play related music is very good."* and *"[...] Love the ability to automatically play similar music. [...]"*.

7. CONCLUSION

We have proposed a new method that combines usage data and social tags to create a high dimensional Euclidean music similarity space using Probabilistic Latent Semantic Analysis. Our experiments show that the combination of usage data and social tags provides a better similarity measure than embeddings based on only one of them. Using the information provided by the underlying tags, an intuitive meaning can be associated with each point in space. To emphasize the analogies with audio feature spaces, we have introduced the notion of *social audio features*. By defining such features for more than 1.4M artists, our work clearly exceeds the volume of existing approaches. *museek*, a smart music player for Android devices, demonstrates that the resulting coverage is high enough to be useable in end user applications. The application uses the proposed similarity space to facilitate novel ways to browse and find music. In the conducted user study we have found that, while traditional methods remain relevant, the advanced retrieval interfaces have been well accepted. In particular, all the incorporated functionality, namely the similarity aware play modes, the music map, and the personalized tag cloud have been appreciated by the end users. This shows that explorative browsing and descriptive music selection methods are indeed important to satisfy the users' needs. The wide acceptance of the implemented interfaces shows that (1) music players should offer similarity based music retrieval functionality, and (2) that the proposed social audio features are well suited to implement this functionality. In fact, we believe that the versatile nature of our social audio features, together with their ability to reflect the users' perception, offers interesting opportunities for future music applications.

8. REFERENCES

[1] J. Aucouturier and F. Pachet. Music Similarity Measures: What's the Use? In *ISMIR*, 2002.

[2] J. Aucouturier and F. Pachet. Scaling up Music Playlist Generation. In *ICME*, 2002.

[3] J. Aucouturier and F. Pachet. Representing musical genre: A state of the art. *JNMR*, 32(1), 2003.

[4] J. Aucouturier and F. Pachet. Improving timbre similarity: How high is the sky. *JNRSAS*, 1(1), 2004.

[5] D. Bainbridge, S. Cunningham, and J. Downie. How people describe their music information needs: A

grounded theory analysis of music queries. In *ISMIR*, 2003.

[6] F. Bentley, C. J. Metcalf, and G. Harboe. Personal vs. commercial content: the similarities between consumer use of photos and music. In *CHI*, 2006.

[7] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *CMJ*, 28(2), 2004.

[8] L. Bossard, M. Kuhn, and R. Wattenhofer. Visually and acoustically exploring the high-dimensional space of music. In *SocialCom*, 2009.

[9] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: current directions and future challenges. *PROCEEDINGS-IEEE*, 96(4), 2008.

[10] M. R. David Gleich, Leonid Zhukov and K. Lang. The World of Music: SDP layout of high dimensional data. In *InfoVis*, 2005.

[11] J. Donaldson and I. Knopke. Music recommendation mapping and interface based on structural network entropy. In *ICDE Workshops*, 2007.

[12] J. S. Downie and S. J. Cunningham. Toward a theory of music information retrieval queries: System design implications. In *ISMIR*, 2002.

[13] D. P. W. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *ISMIR*, 2002.

[14] J. Foote. Content-based retrieval of music and audio. In *SPIE*, volume 3229, 1997.

[15] J. Frank, T. Lidy, P. Hlavac, and A. Rauber. Map-based music interfaces for mobile devices. In *ACM MM*, 2008.

[16] O. Goussevskaia, M. Kuhn, M. Lorenzi, and R. Wattenhofer. From Web to Map: Exploring the World of Music. In *Web Intelligence*, 2008.

[17] O. Goussevskaia, M. Kuhn, and R. Wattenhofer. Exploring music collections on mobile devices. In *Mobile HCI*, 2008.

[18] O. Hilliges, P. Holzer, R. Klüber, and A. Butz. Audioradar: A metaphorical visualization for the navigation of large music collections. In *Smart Graphics*, 2006.

[19] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1), 2001.

[20] P. Knees, M. Schedl, T. Pohle, and G. Widmer. An Innovative Three-Dimensional User Interface for Exploring Music Collections Enriched with Meta-Information from the Web. In *ACM MM*, 2006.

[21] P. Lamere. Social tagging and music information retrieval. *JNMR*, 37(2), 2008.

[22] E. Law, L. Von Ahn, R. Dannenberg, and M. Crawford. Tagatune: A game for music and sound annotation. In *ISMIR 07*, 2003.

[23] J. H. Lee and J. S. Downie. Survey of music information needs, uses, and seeking behaviours: Preliminary findings. In *ISMIR*, 2004.

[24] S. Leitich and M. Topf. Globe of music - music library visualization using geosom. In *ISMIR*, 2007.

[25] M. Levy and M. Sandler. A Semantic Space for Music Derived from Social Tags. In *ISMIR 07*, 2007.

[26] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 2003.

[27] B. Logan. Content-based playlist generation: Exploratory experiments. In *ISMIR*, 2002.

[28] B. Logan and A. Salomon. A music similarity function based on signal analysis. In *ICME*, 2001.

[29] M. Mandel and D. Ellis. A web-based game for collecting music metadata. *JNMR*, 37(2), 2008.

[30] F. Mörchen, A. Ultsch, M. Nöcker, and C. Stamm. Databionic visualization of music collections according to perceptual distance. In *ISMIR*, 2005.

[31] R. Neumayer, M. Dittenbach, and A. Rauber. PlaySOM and PocketSOMPlayer, Alternative Interfaces to Large Music Collections. In *ISMIR*, 2005.

[32] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *ISMIR*, volume 5, 2005.

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

[34] E. Pampalk, T. Pohle, and G. Widmer. Dynamic playlist generation based on skipping behavior. In *ISMIR*, 2005.

[35] J. Platt, C. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a Gaussian Process Prior for Automatically Generating Music Playlists. *NIPS*, 14, 2002.

[36] T. Pohle, E. Pampalk, and G. Widmer. Generating similarity-based playlists using traveling salesman algorithms. In *DAFx*, 2005.

[37] R. Ragno, C. J. C. Burges, and C. Herley. Inferring similarity between music objects with application to playlist generation. In *MIR*, 2005.

[38] F. V. Rob van Gulik. Visual playlist generation on the artist map. In *ISMIR*, 2005.

[39] M. Slaney and W. White. Similarity based on rating data. In *ISMIR*, 2007.

[40] M. Torrens, P. Hertzog, and J. L. Arcos. Visualizing and exploring personal music libraries. In *ISMIR*, 2004.

[41] E. Tsunoo, G. Tzanetakis, N. Ono, and S. Sagayama. Audio genre classification using percussive pattern clustering combined with timbral features. In *ICME*, 2009.

[42] D. Turnbull, L. Barrington, and G. R. G. Lanckriet. Five approaches to collecting tags for music. In *ISMIR*, 2008.

[43] G. Tzanetakis and P. R. Cook. Musical genre classification of audio signals. *IEEE T-SAP*, 10(5), 2002.

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

[45] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. In *ICMC*, 2002.