Context-based Music Similarity Estimation

Markus Schedl and Peter Knees

Johannes Kepler University Linz
Department of Computational Perception
{markus.schedl,peter.knees}@jku.at
http://www.cp.jku.at

Abstract. This review article presents the state-of-the-art in context-based music similarity estimation. It gives an overview of different sources of context-based data on music entities and summarizes various approaches for constructing similarity measures based on the collaborative or cultural knowledge that is incorporated in these data sources. The strength of such context-based measures is elaborated as well as their drawbacks discussed.

1 Motivation

Assessing the similarity of music, musical artists, or musical styles is a non-trivial (one may even say ill-defined or impossible) task. Obviously, there is no explicit definition of what makes two musical entities similar. Is it the melody, the instrumentation, the tempo, or the fact that two artists share certain political views? Answering this question is definitely beyond the scope of this paper, but it is clear that human perception of music similarity depends on several factors – not all of them strongly related to musical properties. Hence, when trying to model music similarity algorithmically, the *cultural context* should not be neglected. This paper tries to give an overview on current methods that aim at capturing some of these cultural aspects of similarity from a variety of different sources.
In the beginnings of music information retrieval (MIR), research on music similarity and related concepts has been focusing on taking into account the symbolic representation of a piece of music, usually given as a MIDI file. As computing power and storage capacities increased, it became feasible to apply and refine signal processing techniques in order to capture certain aspects of a given audio signal, e.g., the rhythmical structure, the aggressiveness, or the timbral shape of a piece of music (for a comprehensive overview of such content-based music information extraction techniques see, e.g., [13]).
In the early 2000s, context-based sources[1] started to be considered an alternative

[1] Although only fuzzily defined in the literature, where the terms "cultural features", "community metadata", and "context-based feature" are commonly used interchangeably, in this paper, we will solely use the term "context-based features" to denote (derived) information on a music entity which is not encoded in some way in the audio file itself, but rather originates from external sources.

for retrieving information on music [26]. Since then, context-based approaches have proven to be very useful for manifold application areas. Context-based information permits, for example, enriching music players with meta-information [56], automatic tagging of artists [18], automatic biography generation [1], or developing user interfaces to browse music collections by more sophisticated means than the textual browsing facilities (in an `artist - album - track` hierarchy) traditionally offered [51, 35]. In this paper, we will focus on the use on context-based features for defining similarity measures between artists and tracks. Music similarity measures can be used, for example, to create relationship networks [12], for automatic playlist generation [6, 52], or to build music recommender systems [15, 64] or music search engines [34]. Furthermore, content-based and context-based features can be beneficially combined in order to ameliorate common MIR tasks, for example, accelerate the creation of playlists [33] or improve the quality of classification according to certain metadata categories like genre, instrument, mood, or listening situation [7].
In the remainder of this paper, we review approaches from the literature that estimate similarity of musical entities from a diversified set of potential sources – from radio station playlists to P2P usage statistics to song lyrics. We describe how these sources are mined in order to construct meaningful features and how these features are then used to create similarity measures. We also try to estimate potential and capabilities of the presented approaches based on the reported evaluations. However, a direct comparison of their performances is not possible, since evaluation strategies and datasets differ largely. Eventually, Section 3 summarizes this work and gives an outlook to possible directions for further research on context-based music information extraction and similarity calculation.

2 Approaches to Context-based Similarity Estimation

In the following, an overview of different context-based approaches to derive music similarity information is given. The approaches found in the literature are categorized by the data source they make use of. A short summary of each method, of the authors' main findings, and of the evaluation results where possible is given.

2.1 Playlists

One of the first approaches to derive similarity information based on the context of a music entity can be found in [49], where *radio station playlists* (extracted from a French radio station) and *compilation CD databases* (using *CDDB*[2]) are exploited to extract co-occurrences between tracks and between artists. The

[2] CDDB is a Web-based album identification service that returns, for a given unique disc identifier, metadata like artist and album name, tracklist, or release year. This service is offered in a commercial version operated by *Gracenote* [25] as well as in an open source implementation named *freeDB* [23].

authors count the number of co-occurrences of two artists (or pieces of music) A_i and A_j on the radio station playlists and compilation CDs. They define the co-occurrence of an entity A_i to itself as the number of occurrences of A_i in the considered corpus. Accounting for different frequencies, i.e., popularity of a song or an artist, is performed by normalizing the co-occurrences. Further, assuming that co-occurrence is a symmetric function, the complete co-occurrence-based similarity measure used by the authors is given in Equation 1.

$$sim_{pl\_cooc}(A_i, A_j) = \frac{1}{2} \cdot \left[\frac{cooc(A_i, A_j)}{cooc(A_i, A_i)} + \frac{cooc(A_j, A_i)}{cooc(A_j, A_j)} \right] \tag{1}$$

However, this similarity measure cannot capture indirect links that an entity may have with others. In order to capture such indirect links, the complete co-occurrence vectors of two entities A_1 and A_2 (i.e., a vector that gives, for a specific entity, the co-occurrence count with all other entities in the corpus) are considered and their statistical correlation is computed, cf. Equation 2.

$$sim_{pl\_corr}(A_i, A_j) = \frac{Cov(A_i, A_j)}{\sqrt{Cov(A_i, A_i) \cdot Cov(A_j, A_j)}} \tag{2}$$

These co-occurrence and correlation functions are used as similarity measures on the track level and on the artist level. Pachet et al. evaluated them on rather small data sets (a set of 12 tracks and a set of 100 artists) using similarity judgments by music experts from *Sony Music* as ground truth. The main finding was that artists or tracks that appear consecutively in radio station playlists or on CD samplers indeed show a high similarity. The co-occurrence function generally performed better than the correlation function (70%–76% vs. 53%-59% agreement with ground truth).

Another work that uses playlists in the context of music similarity estimation is [12]. Cano and Koppenberger created a similarity network via extracting playlist co-occurrences of more than 48,000 artists retrieved from *Art of the Mix* [5] in early 2003. Art of the Mix is a Web service that allows users to upload and share their mixed tapes or playlists. The authors analyzed a total of more than 29,000 playlists. They subsequently created a similarity network where a connection between two artists has been made if they co-occured in a playlist.

The paper reveals some interesting properties of the artist similarity network under consideration. First, each artist is only connected with a small number of other artists. Thus, we can infer that a similarity measure constructed of such data would only capture (strong) positive similarity between two artists. In spite of this sparsity, the network showed one large cluster of nodes connecting more than 99% of the artists. Furthermore, the average shortest path between two artists is remarkably small (3.8). So is the clustering coefficient that estimates the probability of indirect links, i.e., the probability that two neighboring artists of a given one are connected themselves. Thus, given that artist A_1 is similar to A_2 and to A_3, the probability for A_2 and A_3 being similar is quite small (0.1). Analyzing the average degree of a node showed that each artist was on average connected to 12.5 other artists. Since the paper focuses on the network properties, the authors did not perform any other evaluation.

A more recent paper that exploits playlists to derive artist similarity information is [9], where Baccigalupo et al. analyzed co-occurrences of artists in playlists shared by members of a Web community. The authors looked at more than 1 million playlists made publicly available by *MusicStrands* [46], a Web service (no longer in operation) that allows users to share playlists. The authors extracted from the whole playlist set the 4,000 most popular artists, measuring the popularity as the number of playlists in which each artist occurred. They further take into account that two artists that consecutively occur in a playlist are probably more similar than two artists that occur farther away in a playlist. To this end, the authors define a distance function $d_h(A_i, A_j)$ that counts how often a song by artist A_i co-occurs with a song by A_j at a distance of h. Thus, h is a parameter that defines the number of songs in between the occurrence of a song by A_i and the occurrence of a song by A_j in the same playlist. Baccigalupo et al. define the distance between two artists A_i and A_j as in Equation 3, where the playlist counts at distances 0 (two consecutive songs by artists A_i and A_j), 1, and 2 are weighted with β_0, β_1, and β_2, respectively. The authors empirically set the values to $\beta_0 = 1$, $\beta_1 = 0.8$, $\beta_2 = 0.64$.

$$dist_{pl\_d}(A_i, A_j) = \sum_{h=0}^{2} \beta_h \cdot [d_h(A_i, A_j) + d_h(A_j, A_i)] \tag{3}$$

To account for the popularity bias, i.e., very popular artists co-occur with a lot of other artists in many playlists, hence creating a higher similarity to all other artists when simply relying on Equation 3, the authors perform normalization according to Equation 4, where $\widehat{dist_{pl\_d}(A_i)}$ denotes the average distance between A_i and all other artists, i.e., $\frac{1}{n-1} \cdot \sum_{j \in X} dist_{pl\_d}(A_i, A_j)$, and X the set of $n-1$ artists other than A_i.

$$dist_{|pl\_d|}(A_i, A_j) = \frac{dist_{pl\_d}(A_i, A_j) - \widehat{dist_{pl\_d}(A_i)}}{\left| max\left(dist_{pl\_d}(A_i, A_j) - \widehat{dist_{pl\_d}(A_i)}\right)\right|} \tag{4}$$

Unfortunately, no evaluation dedicated to artist similarity was conducted.

2.2 Term Profiles

Another source for cultural features, possibly the most extensive one, is the zillions of available Web pages. Probably one of the earliest works that employs Web mining techniques in the context of MIR can be found in [16]. Cohen and Fan applied collaborative filtering techniques on lists extracted from Web pages. They queried *Altavista* [2] and *Northern Light*[3] [48] to obtain Web pages related

[3] Northern Light, formerly providing a meta search engine, in the meantime has specialized on search solutions tailored to enterprises.

to music artists. The results were then used for artist recommendation. Unfortunately, the paper gives very few details on the exact approach. As ground truth for evaluating their approach, Cohen and Fan exploited server logs of downloads from an internal digital music repository made available within the Intranet of *AT&T*. They analyzed the network traffic for three months, yielding a total of 5,095 artist-related downloads.

In [61] Whitman and Lawrence extracted different term sets (unigrams, bigrams, noun phrases, artist names, and adjectives) from artist-related Web pages found by a search engine. Up to 50 pages that were ranked highest by the search engine were analyzed. After having downloaded the Web pages, the authors applied parsers and a part-of-speech tagger to determine the appropriate term set. Based on term occurrences, individual term profiles were then created for each artist. To this end, the authors employed a simple version of the well-established TF·IDF measure, e.g., [65], that assigns a weight to each term t in the context of each artist A_i. Equation 5 shows the weighting used by the authors, where the term frequency $tf(t, A_i)$ was defined as the percentage of retrieved pages for artist A_i containing term t, and the document frequency $df(t)$ was defined as the percentage of artists (in the whole collection) who had at least one Web page mentioning term t.

$$w_{simple}(t, A_i) = \frac{tf(t, A_i)}{df(t)} \tag{5}$$

Calculating the TF·IDF weights for all terms in each term set yields individual feature vectors or term profiles for each artist. The overlap between the term profiles of two artists was then used as an estimate for their similarity. For evaluation, the authors compared these similarities to two other sources of artist similarity information, which served as ground truth (similar-artist-relations from the online music information system *All Music Guide* (AMG) [4] and user collections from OpenNap, cf. Section 2.5). Remarkable differences between the individual term sets could be made out. The unigram, bigram, and noun phrase sets performed considerably better than the other two sets, regardless of the utilized ground truth definition.

Extending the work presented in [61], Baumann and Hummel [10] introduced certain filters to prune the set of retrieved Web pages. First, they discarded all Web pages with a size of more than 40kB after parsing. They further ignored text in table cells if it did not comprise at least one sentence and more than 60 characters. This should discard adverisements according to the authors. Finally, they performed keyword spotting in the URL, the title, and the first text part of each page. Each occurrence of the words “music”, “review”, and the artist name contributed to a page score. Pages that scored too low were filtered out. In contrast to [61], Baumann and Hummel used a logarithmic weighting of the IDF-term in their TF·IDF formulation. Using these modifications, the authors were able to outperform the approach presented in [61].

Another approach that applies Web mining techniques similarly to [61] is presented in [32]. Knees et al. however do not use specific term sets, but create a term list directly from the retrieved Web pages and use the χ^2-test [63] for term selection, i.e., to filter out terms that are less important to describe certain genres. For similarity computation, this information is a priori unknown. After this step, a variant of the TF·IDF measure was employed to create a weighted term profile for each artist. Equation 6 shows the TF·IDF formulation, where n is the total number of Web pages retrieved for all artists in the collection, $tf(t, A_i)$ is the number of occurrences of term t in all Web pages retrieved for artist A_i, and $df(t)$ is the number of pages in which t occurs at least once.

$$w_{ltc}(t, A_i) = \begin{cases} (1 + \log_2 tf(t, A_i)) \cdot \log_2 \frac{n}{df(t)} & \text{if } tf(t, A_i) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

To calculate the similarity between the term profiles of two artists A_i and A_j, the authors used the cosine similarity according to Formula 7, where T denotes the term set. In this formula, θ gives the angle between A_i's and A_j's feature vectors in the Euclidean space.

$$sim_{cos}(A_i, A_j) = \cos\theta = \left(\frac{\sum\limits_{t \in T} w(t, A_i) \cdot w(t, A_j)}{\sqrt{\sum\limits_{t \in T} w(t, A_i)^2} \cdot \sqrt{\sum\limits_{t \in T} w(t, A_j)^2}} \right) \tag{7}$$

The approach was evaluated in a genre classification setting using k-Nearest Neighbor (k-NN) classifiers on a test collection of 224 artists (14 genres, 16 artists per genre). Accuracies of up to 77% were reported using the k-NN classifier (without term selection).
In [50] similar work is presented. The work of Pampalk et al. however focuses on clustering artists according to TF· IDF feature representations, calculated as in [32]. The authors of [50] manually assembled a dictionary of about 1,400 terms related to music (e.g., genre and style names, instruments, moods, countries) and used this dictionary for term selection. For evaluation, again the 224-artist-set was used. One of the main finding was that considering only the terms in the dictionary when building the feature vectors outperformed using all terms, when the task is to describe artists or clusters of artists. However, when it comes to genre classification using a 1-NN classifier (performing leave-one-out cross validation), the unpruned term set outperformed the use of the dictionary (79% vs. 85% accuracy).

Another approach that extracts TF·IDF features from artist-related Web pages is presented in [53]. Pohle et al. compiled a data set of 1979 artists, which they extracted from AMG. The TF·IDF vectors were calculated for a set of about 3,000 tags extracted from *Last.fm* [36]. The set of tags was constructed by merging tags retrieved for the artists in the collection with Last.fm's most popular tags. For evaluation, k-NN classification experiments with leave-one-out cross validation were performed, resulting in accuracies of about 90%.

There further exist some other approaches that derive term profiles from more specific Web resources. For example, in [14] Celma et al. propose a music search engine that crawls audio blogs via RSS feeds and calculates TF·IDF vectors. Hu et al. in [29] extract TF-based features from music reviews gathered from *Epinions.com* [21].

2.3 Collaborative Tags

As one of the characteristics of the so-called "Web 2.0" – where Web sites encourage (even require) their users to participate in the generation of content – available items such as photos, films, or music can be labeled by the user community with tags. A tag can virtually be anything, but typically it consists of a short description of one aspect typical to the item (for music, for example, genre or style, instrumentation, mood, or performer). The more people are labeling an item with a tag, the more the tag is assumed to be relevant to the item. For music, the most prominent platform that makes use of this option is Last.fm. Since Last.fm provides the collected tags in a standardized manner, it is a very valuable source for context-related information.

In [24] Geleijnse et al. use tags from Last.fm to generate a "tag ground truth" for artists by filtering redundant and noisy tags with the help of tags associated with tracks by the artist under consideration. Similarities between artists are calculated via the number of overlapping tags. Evaluation against Last.fm's similar artist function shows that the number of overlapping tags between similar artists is much larger than the average overlap between arbitrary artists (about 10 vs. 4 after filtering).

In [39] Levy and Sandler retrieved tags from Last.fm and MusicStrands to construct a semantic space for music pieces. To this end, all tags found for a specific track are tokenized like normal text descriptions and a standard TF·IDF-based document-term matrix is created, i.e., each track is represented by a term vector. For the TF, three different calculation methods were explored, namely weighting of the TF by the number of users that applied the tag, no weighting, and restriction to adjectives. Optionally, the dimensionality of the vectors is reduced by applying Latent Semantic Analysis (LSA) [17]. The similarity between vectors is calculating via the cosine measure, cf. Equation 7. For evaluation, for each genre or artist term, each track labeled with that term serves as query, and the mean average precision over all queries is calculated. It is shown that filtering for adjectives clearly worsens the performance of the approach and that weighting of term frequency by the number of users may improve genre precision (however, it is noted that this may just artificially emphasize the majority's opinion without really improving the features). Without LSA (i.e., using the full term vectors) genre precision reaches 80%, and artist precision 61%. Using LSA, genre precision reaches up to 82%, and artist precision 63%. The approach is also compared to the Web-based term profile approach by Knees et al. [32] – cf. Section 2.2. Using the full term vectors in a 1-NN leave-one-out cross validation setting, genre classification rate touches 95% without and 83% with artist filtering.

In comparison to Web-based term approaches, the tag-based approach exhibits some advantages, namely a more music-targeted and smaller vocabulary with significantly less noisy terms and availability of descriptors for individual tracks rather than just artists. On the other hand, tag-based approaches also suffer from some limitations. For example, for sufficient tagging of comprehensive collections, a large and active user community is needed. Furthermore, tagging of tracks from the so-called "long tail", i.e., lesser known tracks, is usually very sparse. Additionally, also effects such as a "community bias" may be observed. To remedy some of these problems, recently, the idea of gathering tags via games has arisen [60, 44, 38]. Such games provide some form of incentive – be it just the pure joy of gaming – to the human player to solve problems that are hard to solve for computers, e.g., capturing emotions evoked when listening to a song. By encouraging users to play such games, a large number of songs can be efficiently annotated with semantic descriptors. Another recent trend to alleviate the data sparsity problem is automatic tagging/propagation of tags based on alternative data sources [59, 19, 30].

2.4 Page Counts and Web Co-Occurrences

This category of approaches analyzes co-occurrences of music entities – usually only the artist level is considered – either on arbitrary Web pages or on specific platforms or services and defines a similarity measure based on such co-occurrence information. For example, in [64] Zadel and Fujinaga investigate the usability of two Web services to derive information on artist similarity. More precisely, they propose an approach that, given a seed artist, retrieves a list of potentially related artists from the *Amazon* [3] Web service *Listmania!*. Based on this list, artist co-occurrences are derived by querying the *Google Web API*[4] and storing the returned page counts of artist-specific queries. Google was queried for `"artist name i"` and for `"artist name i"+"artist name j"`. Thereafter, the so-called "relatedness" of each Listmania! artist to the seed artist is calculated as the ratio between the combined page count, i.e., the number of Web pages on which both artists co-occur, and the minimum of the single page counts of both artists, cf. Equation 8. The minimum is used to account for different popularities of the two artists.

$$sim_{pc\_min}(A_i, A_j) = \frac{pc(A_i, A_j)}{\min\left(pc(A_i), pc(A_j)\right)} \tag{8}$$

Recursively performing the artist extraction from Listmania! and estimating the relatedness to the seed artist via Google page counts allows to construct lists of similar artists. Although the paper has shown that Web services can be used to find similar artists to a seed artist, it lacks a thorough evaluation of the results.

[4] Google no longer offers this Web API. It has been replaced by several other APIs, mostly devoted to Web 2.0 development.

Analyzing Google page counts as a result of artist-related queries was also performed in [55]. Unlike the method presented in [64], Schedl et al. derive complete similarity matrices from artist co-occurrences. This offers additional information since it can also be predicted which artists are **not** similar.
The authors of [55] define the similarity of two artists as the conditional probability that one artist is to be found on a Web page that is known to mention the other artist. Since the retrieved page counts for queries like `"`*`artist name i`*`"` or `"`*`artist name i`*`"+"`*`artist name j`*`"` reveal the relative frequencies of this event, they are used to estimate the conditional probability. Equation 9 gives a more formal representation of the symmetrized similarity function.

$$sim_{pc\_cp}(A_i, A_j) = \frac{1}{2} \cdot \left(\frac{pc(A_i, A_j)}{pc(A_i)} + \frac{pc(A_i, A_j)}{pc(A_j)} \right) \tag{9}$$

In order to restrict the search to Web pages relevant to music, different query schemes were used in [55]. Otherwise, artists that equal common speech words, like “Hole” or “Kiss”, would unjustifiably lead to high page counts, hence, distort the similarity relations. To mitigate this problem, keywords like “music” or “review” were added to the search queries, as already proposed in [61].
Schedl et al. performed two evaluation experiments on the same 224-artist-dataset as used in [32]. They estimated the homogeneity of the genres defined by the ground truth by applying the similarity function to artists within the same genre and to artists from different genres. To this end, the authors related the average similarity between two arbitrary artists from the same genre to the average similarity of two artists from different genres. The results show that the co-occurrence approach can be used to clearly distinguish between most of the genres. The second evaluation experiment was an artist-to-genre classification task using a k-NN classifier. In this setting, the approach yielded in the best case (when combining different query schemes) an accuracy of about 85% averaged over all genres.

Unlike the approaches that create term profiles from Web pages, co-occurrence analysis only makes use of the page counts. Therefore, Web traffic can be minimized by restricting the search to display only the top-ranked page if the used search engine offers such an option. However, a shortcoming of these approaches is that creating a complete similarity matrix has quadratic computational complexity in the number of artists. It therefore scales poorly as the number of queries that has to be issued to the search engine grows quadratically with the number of artists in the collection.
The quadratic computational complexity can be avoided by employing another strategy to co-occurrence analysis as described in [54][Chapter 3]. In a first step, for each artist A_i, a certain amount of top-ranked Web pages returned by the search engine is retrieved. Subsequently, all pages fetched for artist A_i are searched for occurrences of all other artist names A_j in the collection. The number of page hits again represents a co-occurrence count or a document frequency of the artist term “A_j” in the corpus given by the Web pages for artist

A_i. Relating this count to the total number of pages successfully fetched for artist A_i, a similarity function can be constructed that requires the number of issued queries only to be equal to the number of artists in the collection. The formula for the symmetric artist similarity equals Equation 1.

2.5 Peer-to-Peer Network Co-Occurrences

Peer-to-peer (P2P) networks represent a rich source for mining music-related data since their users are commonly willing to reveal various kinds of metadata about the shared content. In the case of shared music files, file names and ID3 tags are usually disclosed.

Early work that makes use of data extracted from P2P networks comprises [61], [20], [40], and [11]. All these papers use, among other sources, data extracted from the P2P network *OpenNap* to derive music similarity information. Although it is unclear whether the four publications make use of exactly the same data set, the respective authors all state that they extracted metadata, but did not download any files, from OpenNap. [40] and [11] report on having determined the 400 most popular artists on OpenNap in mid 2002. The authors gathered metadata on shared content, which yielded about 175,000 user-to-artist relations from about 3,200 shared music collections. [40] especially highlights the sparsity in the OpenNap data, in comparison with data extracted from the audio signal. Although this is obviously true, the authors miss to note the inherent disadvantage of signal-based feature extraction, i.e., extracting signal-based features is only possible when the audio content is available. Logan et al. then compared similarities defined by artist co-occurrences in OpenNap collections, by expert opinions from AMG, by playlist co-occurrences from Art of the Mix, by data gathered from a Web survey, and by audio feature extracion via MFCCs, e.g. [8]. To this end, they calculated a "ranking agreement score", which is basically comparing the top N most similar artists according to each data source and calculating the pair-wise overlap between the sources. The main findings were that the co-occurrence data from OpenNap and from Art of the Mix showed a high degree of overlap, the experts from AMG and the participants of the Web survey showed a moderate agreement, and the signal-based measure had a rather low agreement with all other sources (except when compared it with the AMG data). In [61] a software agent was used to retrieve from OpenNap a total of 1.6 million user-song entries over a period of three weeks in August 2001. To alleviate the popularity bias of the data, Whitman and Lawrence used a similarity measure as shown in Equation 10, where $C(A_i)$ denotes the number of users that share songs by artist A_i, $C(A_i, A_j)$ is the number of users that have both artists A_i and A_j in their shared collection, and A_k is the most popular artist in the corpus. The right term in the equation downweights the similarity between two artists if one of them is very popular and the other not.

$$sim_{p2p\_wl}(A_i, A_j) = \frac{C(A_i, A_j)}{C(A_j)} \cdot \left(1 - \frac{|C(A_i) - C(A_j)|}{C(A_k)}\right) \tag{10}$$

In [20] Ellis et al. use the same artist set as in [61]. The aim is to build a ground truth for artist similarity estimation. They report on having extracted from OpenNap about 400,000 user-to-song relations, covering about 3,000 unique artists. Again, the co-occurrence data is compared with artist similarity data gathered by a Web survey and with AMG data. In contrast to [61], [20] take indirect links in AMG's similarity judgments into account. To this end, Ellis et al. propose a transitive similarity function on similar artists from the AMG data, which they call "Erdös distance". More precisely, the distance $d(A_1, A_2)$ between two artists A_1 and A_2 is measured as the minimum number of intermediate artists needed to form a path from A_1 to A_2. As this procedure also allows to derive information on dissimilar artists (those with a high minimum path length), it can be employed to obtain a complete distance matrix. Furthermore, the authors propose an adapted distance measure, the so-called "Resistive Erdös measure", which takes into account that there may exist more than one shortest path of length l between A_1 and A_2. Assuming that two artists are more similar if they are connected via many different paths of length l, the Resistive Erdös similarity measure equals the electrical resistance in a network, cf. Equation 11, where each path from A_i to A_j is modeled as a resistor whose resistance equals the path length $|p|$. However, this adjustment did not improve the agreement of the similarity measure with the data from the Web-based survey, as it failed to overcome the popularity bias, i.e., many different paths between popular artists unjustifiably lower the total resistance.

$$dist_{p2p\_res}(A_i, A_j) = \frac{1}{\displaystyle\sum_{p \in Paths(A_i, A_j)} \frac{1}{|p|}} \tag{11}$$

A recent approach that derives similarity information on the artist and on the song level from the *Gnutella* P2P file sharing network is presented in [57]. Shavitt and Weinsberg collected metadata of shared files from more than 1.2 million Gnutella users in November 2007. They restricted their search to music files (.mp3 and .wav). The crawl yielded a data set of 530,000 songs. Information on both users and songs were then represented via a 2-mode graph showing users and songs. A link between a song and a user was created when the user shared the song. One finding of analyzing the resulting network was that most users in the P2P network shared similar files.

The authors used the data gathered for artist recommendation. To this end, they constructed a user-to-artist matrix V, where $V(i, j)$ gives the number of songs by artist A_j that user U_i shared. Shavitt and Weinsberg then performed direct clustering on V using the k-means algorithm [42] with the Euclidean distance metric. Artist recommendation is then performed using either data from the centroid of the cluster to which the seed user U_i belongs or by using the nearest neighbors of U_i within the cluster to which U_i belongs.

In addition, Shavitt and Weinsberg also addressed the problem of song clustering. Accounting for the popularity bias, the authors defined a distance function that is normalized according to song popularity, as shown in Equation 12, where $uc(S_i, S_j)$ denotes the total number of users that share songs S_i and S_j, and C_i and C_j denote, respectively, the popularity of songs S_i and S_j, measured as their total occurrence in the corpus.

$$dist_{p2p\_pop}(S_i, S_j) = -log_2 \left(\frac{uc(S_i, S_j)}{\sqrt{C_i \cdot C_j}} \right) \tag{12}$$

Evaluation experiments were carried out for song clustering. The authors reported an average precision of 12.1% and an average recall of 12.7%, which they judged as quite good when considering the vast amount of songs shared by the users and the inconsistency in the metadata (ID3 tags).

2.6 Song Lyrics

The lyrics of a song represent an important aspect of the semantics of music since they usually reveal information about the artist or the performer: e.g., cultural background (via different languages or use of slang words), political orientation, or style of music (use of a specific vocabulary in certain music styles).
Logan et al. use song lyrics for tracks by 399 artists to determine artist similarity [41]. To this end, in a first step, Probabilistic Latent Semantic Analysis (PLSA) [27] is applied to a collection of over 40,000 song lyrics to extract N topics typical to lyrics. In a second step, all lyrics by an artist are processed using each of the extracted topic models to create N-dimensional vectors of which each dimension gives the likelihood of the artist's tracks to belong to the corresponding topic. Artist vectors are then compared by calculating the L_1 distance (also known as Manhattan distance) as shown in Equation 13.

$$dist_{L_1}(A_i, A_j) = \sum_{k=1}^{N} |a_{i,k} - a_{j,k}| \tag{13}$$

This similarity approach is evaluated against human similarity judgments, i.e., the "survey" data for the *uspop2002* set [11], and yields worse results than similarity data obtained via acoustic features (irrespective of the chosen N, the usage of stemming, or the filtering of lyrics-specific stopwords). However, as lyrics-based and audio-based approaches make different errors, a combination of both is suggested. In [43] Mahedero et al. demonstrate the usefulness of lyrics for four important tasks: language identification, structure extraction (i.e., recognition of intro, verse, chorus, bridge, outro, etc.), thematic categorization, and similarity measurement. For similarity calculation, a standard TF·IDF measure with cosine distance is proposed as initial step. Using this information, a song's representation is obtained by concatenating distances to all songs in the collection into a new vector. These representations are then compared using an unspecified algorithm. Exploratory experiments indicate some potential for cover version identification and plagiarism detection.

Other approaches are not explicitly aiming at finding similar songs in terms of lyrical (or rather semantic) content but at revealing conceptual clusters [31] or to classify songs into genres [45] or mood categories [37, 28]. However, most of these approaches are nevertheless of interest to us, as extracted features can in principle also be used for similarity calculation. In [37], the goal of Laurier et al. is classification of songs to four mood categories by means of lyrics and content analysis. For lyrics, the TF·IDF measure with cosine distance is incorporated. Optionally, also LSA is applied to the TF·IDF vectors (achieving best results when projecting vectors down to 30 dimensions). In both cases, a 10-fold cross validation with k-NN classification yielded accuracies slightly above 60%. Audio-based features performed better compared to lyrics features, however, a combination of both yielded best results. Hu et al. experiment with TF·IDF, TF, and Boolean vectors and investigate the impact of stemming, part-of-speech tagging, and function words for soft-categorization into 18 mood clusters [28]. Best results are achieved with TF·IDF weights on stemmed terms. An interesting result is that in this scenario, lyrics-based features alone can outperform audio-based features. Beside TF·IDF and part-of-speech features, Mayer et al. [45] also propose the use of rhyme and statistical features to improve lyrics-based genre classification. To extract rhyme features, lyrics are transcribed to a phonetic representation and searched for different patterns of rhyming lines (e.g., AA, AABB, ABAB). Features consist of the number of occurrence of each pattern, as well as the percentage of rhyming blocks and the fraction of unique terms used to build the rhymes. Statistical features are constructed by counting various punctuation characters and digits and calculating typical ratios like average words per line or average length of words. Classification experiments show that the proposed style features and also a combination of style features and classical TF·IDF features outperforms the TF·IDF only approach.

In summary, recent literature demonstrates that many interesting aspects of context-based similarity can be covered by exploiting lyrics information. However, since new and ground breaking applications for this kind of information have yet not been discovered, the potential of lyrics analysis is currently mainly seen as a complementary source to content-based features for genre or mood classification.

3 Discussion and Outlook

In this paper, we have given an overview of approaches to estimate music similarity that do not rely on the audio signal, but rather take various aspects of the context in which a music entity occurs into consideration.

Even though the presented context-based approaches demonstrate the great potential of comprehensive community data, basically all of them suffer from similar shortcomings. First, *data sparsity*, especially for artists in the "long tail", is obviously a problem. Second, the *popularity bias* has to be addressed, i.e., that disproportionately more data is available for popular artists than for lesser known ones, which often distorts derived similarity measures. Furthermore, methods that aim at milking user-based data are prone to include only participants of existing communities in a broad sense (from very specific services, like a certain P2P network, to the Web community as a whole). It is further known that users of certain communities tend to have similar music tastes. In general, this phenomenon is known as *community* or *population bias*; in the case of Last.fm, we would suggest the term "Radiohead bias" instead.

For the future, we believe that it is crucial to transcend the idea of a generally valid notion of similarity and establish a differentiated, multi-granular concept of similarity (that takes into account regional particularities and views and adapts to cultural areas as well as to individuals). This becomes particularly apparent when comparing current representations of Western and non-Western music. Furthermore, we think that multi-faceted similarity measures will be standard in music applications. They may be defined as a mixture of content- and context-based aspects, e.g., to enable retrieval systems capable of dealing with queries like "give me rhythmically similar music to the most recent chart hits in Canada, but which was released in the 1970s".

In this paper, we focused on the currently most prominent sources to derive context-based music information from. There exist, however, alternative data sources that are considered in the literature. For example, in [58] user ratings of playlists from the *Yahoo!* music service [62] were analyzed (1.5 million judgments by 380,000 users). In [22] Fields et al. propose the usage of artist-related social network data from *MySpace* [47]. The authors state that similarity information based on the artists' "top friends" seems to be a promising complement to signal-based audio similarity.

These are just two examples of potential other sources, and it is guaranteed that there are even more, yet to discover. Since music plays a central role in many people's lives, references to music can be found everywhere. For music research, the big challenge is to discover such sources and make them accessible.

Acknowledgments

This research is supported by the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) under project number L511-N15. Furthermore, we wish to acknowledge all authors whose contributions to context-based music similarity we missed to address here.

References

1. H. Alani, S. Kim, D. E. Millard, M. J. Weal, W. Hall, P. H. Lewis, and N. R. Shadbolt. Automatic Ontology-Based Knowledge Extraction from Web Documents. *IEEE Intelligent Systems*, 18(1):14–21, 2003.
2. http://www.altavista.com (access: February 2008).
3. http://www.amazon.com (access: January 2008).
4. http://www.allmusic.com (access: November 2007).

5. http://www.artofthemix.org (access: February 2008).
6. J.-J. Aucouturier and F. Pachet. Scaling Up Music Playlist Generation. In *Proc IEEE ICME*, 2002.
7. J.-J. Aucouturier, F. Pachet, P. Roy, and A. Beurivé. Signal + Context = Better Classification. In *Proc 8th ISMIR*, 2007.
8. J.-J. Aucouturier, F. Pachet, and M. Sandler. "The Way It Sounds": Timbre Models for Analysis and Retrieval of Music Signals. *IEEE Transactions on Multimedia*, 7(6):1028–1035, Dec 2005.
9. C. Baccigalupo, E. Plaza, and J. Donaldson. Uncovering Affinity of Artists to Multiple Genres from Social Behaviour Data. In *Proc 9th ISMIR*, 2008.
10. S. Baumann and O. Hummel. Using Cultural Metadata for Artist Recommendation. In *Proc 3rd WEDELMUSIC*, 2003.
11. A. Berenzweig, B. Logan, D. P. Ellis, and B. Whitman. A Large-Scale Evaluation of Acoustic and Subjective Music Similarity Measures. In *Proc 4th ISMIR*, 2003.
12. P. Cano and M. Koppenberger. The Emergence of Complex Network Patterns in Music Artist Networks. In *Proc 5th ISMIR*, 2004.
13. M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-Based Music Information Retrieval: Current Directions and Future Challenges. *Proc IEEE*, 96:668–696, Apr 2008.
14. O. Celma, P. Cano, and P. Herrera. SearchSounds: An Audio Crawler Focused on Weblogs. In *Proc 7th ISMIR*, 2006.
15. O. Celma and P. Lamere. ISMIR 2007 Tutorial: Music Recommendation. http://mtg.upf.edu/∼ocelma/MusicRecommendationTutorial-ISMIR2007 (access: December 2007), 2007.
16. W. W. Cohen and W. Fan. Web-Collaborative Filtering: Recommending Music by Crawling The Web. *WWW9 / Computer Networks*, 33(1–6):685–698, 2000.
17. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.
18. D. Eck, T. Bertin-Mahieux, and P. Lamere. Autotagging Music Using Supervised Machine Learning. In *Proc 8th ISMIR*, 2007.
19. D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic Generation of Social Tags for Music Recommendation. In *Advances in Neural Information Processing Systems 20 (NIPS'07)*. MIT Press, 2008.
20. D. P. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The Quest For Ground Truth in Musical Artist Similarity. In *Proc 3rd ISMIR*, 2002.
21. http://www.epinions.com/music (access: August 2007).
22. B. Fields, M. Casey, K. Jacobson, and M. Sandler. Do You Sound Like Your Friends? Exploring Artist Similarity via Artist Social Network Relationships and Audio Signal Processing. In *Proc ICMC*, 2008.
23. http://www.freedb.org (access: February 2008).
24. G. Geleijnse, M. Schedl, and P. Knees. The Quest for Ground Truth in Musical Artist Tagging in the Social Web Era. In *Proc 8th ISMIR*, 2007.
25. http://www.gracenote.com (access: February 2008).
26. M. Grachten, M. Schedl, T. Pohle, and G. Widmer. The ISMIR Cloud: A Decade of ISMIR Conferences at Your Fingertips. In *Proc 10th ISMIR*, 2009.
27. T. Hofmann. Probabilistic Latent Semantic Analysis. In *Proc Uncertainty in Artificial Intelligence (UAI)*, 1999.
28. X. Hu, J. S. Downie, and A. F. Ehmann. Lyric Text Mining in Music Mood Classification. In *Proc 10th ISMIR*, 2009.
29. X. Hu, J. S. Downie, K. West, and A. Ehmann. Mining Music Reviews: Promising Preliminary Results. In *Proc 6th ISMIR*, 2005.
30. J. H. Kim, B. Tomasik, and D. Turnbull. Using Artist Similarity to Propagate Semantic Information. In *Proc 10th ISMIR*, 2009.
31. F. Kleedorfer, P. Knees, and T. Pohle. Oh Oh Oh Whoah! Towards Automatic Topic Detection In Song Lyrics. In *Proc 9th ISMIR*, 2008.
32. P. Knees, E. Pampalk, and G. Widmer. Artist Classification with Web-based Data. In *Proc 5th ISMIR*, 2004.
33. P. Knees, T. Pohle, M. Schedl, and G. Widmer. Combining Audio-based Similarity with Web-based Data to Accelerate Automatic Music Playlist Generation. In *Proc 8th ACM MIR*, 2006.
34. P. Knees, T. Pohle, M. Schedl, and G. Widmer. A Music Search Engine Built upon Audio-based and Web-based Similarity Measures. In *Proc 30th ACM SIGIR*, 2007.

35. 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 *Proc 14th ACM Multimedia*, 2006.
36. http://last.fm (access: December 2007).
37. C. Laurier, J. Grivolla, and P. Herrera. Multimodal Music Mood Classification Using Audio and Lyrics. In *Proc ICMLA*, 2008.
38. E. Law, L. von Ahn, R. Dannenberg, and M. Crawford. Tagatune: A Game for Music and Sound Annotation. In *Proc 8th ISMIR*, 2007.
39. M. Levy and M. Sandler. A Semantic Space for Music Derived from Social Tags. In *Proc 8th ISMIR*, 2007.
40. B. Logan, D. P. Ellis, and A. Berenzweig. Toward Evaluation Techniques for Music Similarity. In *Proc 26th ACM SIGIR: Workshop on the Evaluation of Music Information Retrieval Systems*, 2003.
41. B. Logan, A. Kositsky, and P. Moreno. Semantic Analysis of Song Lyrics. In *Proc IEEE ICME*, 2004.
42. J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In L. M. L. Cam and J. Neyman, editors, *Proc 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.
43. J. P. G. Mahedero, A. Martínez, P. Cano, M. Koppenberger, and F. Gouyon. Natural Language Processing of Lyrics. In *Proc 13th ACM Multimedia*, 2005.
44. M. I. Mandel and D. P. Ellis. A Web-based Game for Collecting Music Metadata. In *Proc 8th ISMIR*, 2007.
45. R. Mayer, R. Neumayer, and A. Rauber. Rhyme and Style Features for Musical Genre Classification by Song Lyrics. In *Proc 9th ISMIR*, 2008.
46. http://music.strands.com (access: November 2009).
47. http://www.myspace.com (access: November 2009).
48. http://www.northernlight.com (access: February 2008).
49. F. Pachet, G. Westerman, and D. Laigre. Musical Data Mining for Electronic Music Distribution. In *Proc 1st WEDELMUSIC*, 2001.
50. E. Pampalk, A. Flexer, and G. Widmer. Hierarchical Organization and Description of Music Collections at the Artist Level. In *Proc 9th ECDL*, 2005.
51. E. Pampalk and M. Goto. MusicSun: A New Approach to Artist Recommendation. In *Proc 8th ISMIR*, 2007.
52. T. Pohle, P. Knees, M. Schedl, E. Pampalk, and G. Widmer. "Reinventing the Wheel": A Novel Approach to Music Player Interfaces. *IEEE Transactions on Multimedia*, 9:567–575, 2007.
53. T. Pohle, P. Knees, M. Schedl, and G. Widmer. Building an Interactive Next-Generation Artist Recommender Based on Automatically Derived High-Level Concepts. In *Proc 5th CBMI*, 2007.
54. M. Schedl. *Automatically Extracting, Analyzing, and Visualizing Information on Music Artists from the World Wide Web*. PhD thesis, Johannes Kepler University, Linz, Austria, 2008.
55. M. Schedl, P. Knees, and G. Widmer. A Web-Based Approach to Assessing Artist Similarity using Co-Occurrences. In *Proc 4th CBMI*, 2005.
56. M. Schedl, T. Pohle, P. Knees, and G. Widmer. Assigning and Visualizing Music Genres by Web-based Co-Occurrence Analysis. In *Proc 7th ISMIR*, 2006.
57. Y. Shavitt and U. Weinsberg. Songs Clustering Using Peer-to-Peer Co-occurrences. In *Proc IEEE ISM: AdMIRe*, 2009.
58. M. Slaney and W. White. Similarity Based on Rating Data. In *Proc 8th ISMIR*, 2007.
59. M. Sordo, C. Laurier, and O. Celma. Annotating Music Collections: How Content-based Similarity Helps to Propagate Labels. In *Proc 8th ISMIR*, 2007.
60. D. Turnbull, R. Liu, L. Barrington, and G. Lanckriet. A Game-based Approach for Collecting Semantic Annotations of Music. In *Proc 8th ISMIR*, 2007.
61. B. Whitman and S. Lawrence. Inferring Descriptions and Similarity for Music from Community Metadata. In *Proc ICMC*, 2002.
62. http://music.yahoo.com (access: November 2007).
63. Y. Yang and J. O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *Proc 14th ICML*, 1997.
64. M. Zadel and I. Fujinaga. Web Services for Music Information Retrieval. In *Proc 5th ISMIR*, 2004.
65. J. Zobel and A. Moffat. Exploring the Similarity Space. *ACM SIGIR Forum*, 32(1):18–34, 1998.