Available online at www.sciencedirect.com
SciVerse ScienceDirect
journal homepage: www.elsevier.com/locate/cosrev

Survey

Contextual music information retrieval and recommendation: State of the art and challenges

*Marius Kaminskas**, *Francesco Ricci*

Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy

ARTICLE INFO

Article history:
Received 15 September 2011
Received in revised form
30 March 2012
Accepted 7 April 2012

Keywords:
Music information retrieval
Music recommender systems
Context-aware services
Affective computing
Social computing

ABSTRACT

Increasing amount of online music content has opened new opportunities for implementing new effective information access services – commonly known as music recommender systems – that support music navigation, discovery, sharing, and formation of user communities. In the recent years a new research area of contextual (or situational) music recommendation and retrieval has emerged. The basic idea is to retrieve and suggest music depending on the user's actual situation, for instance emotional state, or any other contextual conditions that might influence the user's perception of music. Despite the high potential of such idea, the development of real-world applications that retrieve or recommend music depending on the user's context is still in its early stages. This survey illustrates various tools and techniques that can be used for addressing the research challenges posed by context-aware music retrieval and recommendation. This survey covers a broad range of topics, starting from classical music information retrieval (MIR) and recommender system (RS) techniques, and then focusing on context-aware music applications as well as the newer trends of affective and social computing applied to the music domain.

Contents

* *Corresponding author.*
E-mail addresses: mkaminskas@unibz.it (M. Kaminskas), fricci@unibz.it (F. Ricci).

1574-0137/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.cosrev.2012.04.002

1. Introduction

Music has always played a major role in human entertainment. With the coming of digital music and Internet technologies, a huge amount of music content has become available to millions of users around the world. With millions of artists and songs on the market, it is becoming increasingly difficult for the users to search for music content—there is a lot of potentially interesting music that is difficult to discover. Furthermore, huge amounts of available music data have opened new opportunities for researchers working on music information retrieval and recommendation to create new viable services that support music navigation, discovery, sharing, and formation of user communities. The demand for such services – commonly known as music recommender systems – is high due to the economic potential of online music content.

Music recommender systems are decision support tools that reduce the information overload by retrieving only items that are estimated as relevant for the user, based on the user's profile, i.e., a representation of the user's music preferences [1]. For example, Last.fm[1] – a popular Internet radio and recommender system – allows a user to mark songs or artists as favorites. It also tracks the user's listening habits, and based on this information can identify and recommend music content that is more likely to be interesting to the user.

However, most of the available music recommender systems suggest music without taking into consideration the user's context, e.g., her mood, or her current location and activity [2]. In fact, a study on users' musical information needs [3] showed that people often seek music for a certain occasion, event, or an emotional state. Moreover, the authors of a similar study [4] concluded that there is a growing need for "*extra-musical information*" that would "*contextualize users' real-world searches*" for music to provide more useful retrieval results.

In response to these observations, in recent years a new research topic of contextual (or situational) music retrieval and recommendation has emerged. The idea is to recommend music depending on the user's actual situation, e.g., her emotional state, or any other contextual condition that might influence the user's perception or evaluation of music. Such music services can be used in new engaging applications. For instance, location-aware systems can retrieve music content that is relevant to the user's location, e.g., by selecting music composed by artists that lived in that location. Or, a mobile tourist guide application could play music that fits the place the tourist is visiting, by selecting music tracks that match the emotions raised in that place [5]. Or finally, an in-car music player may adapt music to the landscape the car is passing [6].

However, despite the high potential of such applications, the development of real-world context-aware music recommenders is still in its early stages. Few systems are actually released to the market as researchers are facing numerous challenges when developing effective context-aware music delivery systems. The majority of these challenges pertain to the heterogeneity of data, i.e., in addition to dealing with music, researchers must consider various types of

[1] http://www.last.fm.

contextual information (e.g., emotions, time, location, multimedia). Another challenge is related to the high cost of evaluating context-aware systems—the lack of reference datasets and evaluation frameworks makes every evaluation time consuming, and often requires real users' judgments.

In order to help researchers in addressing the above mentioned challenges of context-aware music retrieval and recommendation, we provide here an overview of various topics related to this area. Our main goal is to illustrate the available tools and techniques that can be used for addressing the research challenges. This review covers a broad range of topics, starting from classical music information retrieval (MIR) and recommender system (RS) techniques, and subsequently focusing on context-aware music applications as well as the methods of affective and social computing applied to the music domain.

The rest of this paper is structured as follows. In Section 2 we review the basic techniques of content-based music retrieval. Section 3 provides an overview of the state-of-the-art in the area of recommender systems, their application in the music domain, and describes some of the popular commercial music recommenders. In Section 4 we discuss the newer trends of MIR research—we first discuss the research in the area of contextual music retrieval and recommendation, and describe some prototype systems (Section 4.1). Subsequently, we review the automatic emotion recognition in music (Section 4.2) and the features of Web 2.0 – online communities and social tagging – applied to music domain (Section 4.3). Finally, in Section 5 we present some conclusions of this survey and provide links to the relevant scientific conferences.

2. Content-based music information retrieval

In this section we give an overview of traditional music information retrieval techniques, where audio content analysis is used to retrieve or categorize music. Music information retrieval (MIR) is a part of a larger research area—multimedia information retrieval. Researchers working in this area focus on retrieving information from different types of media content: images, video, and sounds. Although these types of content differ from each other, separate disciplines of multimedia information retrieval share techniques like pattern recognition and learning techniques. This research field was born in the 80's, and initially focused on computer vision [7]. The first research works on audio signal analysis started with automatic speech recognition and discriminating music from speech content [8].

In the following years the field of music information retrieval grew to cover a wide range of techniques for music analysis. For computers (unlike humans), music is nothing else than a form of audio signal. Therefore, MIR uses audio signal analysis to extract meaningful features of music. An overview of information extraction from audio [9] identified three levels of information that can be extracted from a raw audio signal: event-scale information (i.e., transcribing individual notes or chords), phrase-level information (i.e., analyzing note sequences for periodicities), and piece-level information (i.e., analyzing longer excerpts of audio tracks).

While event-scale information can be useful for instrument detection in a song, or for query by example and query by humming (see Sections 2.1 and 2.2), it is not the most salient way to describe music. Phrase-level information analyzes longer temporal excerpts and can be used for tempo detection, playlist sequencing, or music summarization (finding a representative piece of a track). Piece-level information is related to a more abstract representation of a music track, closer to user's perception of music, and therefore can be used for tasks as genre detection, or user preference modeling in content-based music recommenders (see Section 3.2).

A survey of existing MIR systems was presented by Typke et al. [10]. In this work the systems were analyzed with respect to the level of retrieval tasks they perform. The authors defined four levels of retrieval tasks: genre level, artist level, work level, and instance level. For instance: searching for rock songs is a task at a genre level; looking for artists similar to Björk is clearly a task at an artist level; finding cover versions of the song "Let it Be" by The Beatles is a task at a work level; finally, identifying a particular recording of Mahler's fifth symphony is a task at an instance level. The survey concluded that the available systems focus on work/instance and genre levels. The authors identified the lack of systems on the artist level as a gap between specific and general retrieval oriented systems. Interesting MIR applications, like artist analysis or specific music recommendations fall into this gap. The authors suggested it is important to find algorithms for representing music at a higher, more conceptual abstraction level than the level of notes although no specific suggestions were made.

Despite the advances of MIR research, automatic retrieval systems still fail to cover the semantic gap between the language used by humans (information seekers) and computers (information providers). Nowadays, researchers in the field of multimedia IR (and music IR in particular) focus on methods to bring information retrieval closer to humans by means of human-centric and affective computing [7].

In this section we review the traditional applications of music information retrieval—query by example, query by humming, and genre classification.

2.1. *Query by example*

Query by example (QBE) was one of the first applications of MIR techniques. Systems implementing this approach are taking audio signal as an input, and return the metadata information of the recording—artist, title, genre, etc. A QBE system can be useful to users who have access to a recording and want to obtain the metadata information (e.g., finding out which song is playing on the radio, or getting information about an unnamed mp3 file).

QBE uses audio fingerprinting technique [11]. It is a technique for representing a specific audio recording in a unique way (similarly to fingerprints representing humans in a unique way) using the low-level audio features. Such approach is good for identifying a specific recording, not a work in general. For instance, a QBE system would recognize an album version of "Let it Be" by The Beatles, but various live performances or cover versions of the same song most likely would not be recognized due to the differences in the audio signal.

There are two fundamental parts in audio fingerprinting—fingerprint extraction and matching. Fingerprints of audio tracks must be robust, have discrimination power over huge amounts of other fingerprints, and be resistant to distortions.

One of the standard approaches to extract features for audio fingerprinting is calculating the Mel-Frequency Cepstrum Coefficients (MFCCs). MFCCs are spectral-based features that are calculated for short time frames (typically 20 ms) of the audio signal. This approach has been primarily used in speech recognition research, but has been shown to perform well also when modeling music signal [12].

Besides MFCCs, features like spectral flatness, tone peaks, and band energy are also used for audio fingerprinting [11]. Often, derivatives and second order derivatives of signal features are used. The extracted features are typically stored as feature vectors. Given a fingerprint model, a QBE system searches a database of fingerprints for matches. Similarity measures used for matching include Euclidean, Manhattan, and Hamming distances [11].

One of the early QBE methods was developed in 1996 by researchers at Muscle Fish company [13]. Their approach was based on signal features describing loudness, pitch, brightness, bandwidth, and harmonicity. Euclidean distance was used to measure similarity between feature vectors. The approach was designed to recognize short audio samples (i.e., sound effects, speech fragments, single instrument recordings), and is not applicable to complex or noisy audio data.

Nowadays, one of the most popular QBE systems is Shazam music recognition service [14]. It is a system running on mobile devices that records 10 s of audio, performs feature extraction on the mobile device to generate an audio fingerprint, and then sends the fingerprint to Shazam server which performs the search on the database of audio fingerprints and returns the matching metadata.

The fingerprinting algorithm has to be resistant to noise and distortions, since the users can record audio in a bar or on a street. Shazam researchers found that standard features like MFCCs were not robust enough to handle the noise in the signal. Instead, spectrogram peaks – local maximums of the signal frequency curve – were used as the basis for audio fingerprints.

2.2. *Query by humming*

Query by humming (QBH) is an application of MIR techniques that takes an input of a melody sung (or hummed) by the user, and retrieves the matching track and its metadata. QBH systems cannot use the audio fingerprinting techniques of QBE systems since their goal is to recognize altered versions of a song (e.g., a hummed tune or a live performance) that a QBE system would most likely fail to retrieve [15].

As users can only hum melodies that are memorable and recognizable, QBH is only suitable for melodic music, not for rhythmic or timbral compositions (e.g., African folk music). The melody supplied by the user is monophonic. Since most western music is polyphonic, individual melodies must be extracted from the tracks in the database to match them with the query. The standard audio format is not suitable for this task, therefore, MIDI format files are used. Although MIDI files contain separate tracks for each instrument, the perceived melody may be played by multiple instruments, or switch from one instrument to another. A number of approaches to extracting individual melodies from MIDI files have been proposed [16,17].

The MIDI files are prepared in such a way that they represent not entire pieces, but the main melodic themes (e.g., the first notes of Beethoven's fifth symphony). This helps avoiding accidental matches with unimportant parts of songs, since users tend to supply main melodic themes as queries. To extract such main themes is a challenging task, since they can occur anywhere in a track, and can be performed by any instrument. Typically, melodic theme databases are built manually by domain experts, although there are successful attempts to do this automatically [18].

Since in QBH systems the query supplied by the user is typically distant from the actual recording in terms of low-level audio features like MFCCs, these systems must perform matching at a more abstract level, looking for melodic similarity. Melody is related to pitch distribution in audio segments. Therefore, similarity search is based on pitch information. In MIDI files, the features describing music content are: pitch, starting time, duration, and relative loudness of every note. For the hummed query, pitch information is extracted by transcribing audio signal into individual notes [19].

The similarity measures used by different QBH systems depend on the representation of pitch information. When melodies are represented as strings of either absolute or relative pitch values, approximate string matching (string edit distance) is used to find similar melodies. Other approaches represent pitch intervals as *n*-grams, and use the *n*-gram overlap between the query and database items as a similarity measure. Hidden Markov Models (HMM) are also used in query by humming systems, and allow to model the errors that the users make when humming a query [19].

A pioneer QBH system was introduced by Ghias et al. [20]. The authors used a string representation of music content and approximate string matching algorithm to find similar melodies. The system functioned with a database of 183 songs.

In a more recent work, Pardo et al. [21] implemented and compared two approaches to query by humming—the first based on approximate string matching, and the second based on the Hidden Markov Model. The results showed that none of the two approaches is significantly superior to the other. Moreover, neither approach surpassed human performance.

2.3. *Genre classification*

Unlike the previously described applications of music information retrieval, determining the genre of music is not a search, but a classification problem. Assigning genre labels to music tracks is important for organizing large music collections, helping users to navigate and search for music content, create automatic radio stations, etc.

A major challenge for the automatic genre classification task is the fuzziness of the *genre* concept. As of today, there is no defined general taxonomy of music genres. Each of the popular music libraries[2] use their own hierarchy of genres that have little terms in common [22]. Furthermore, music genres are constantly evolving with new genre labels appearing yearly. Since attempts to create a unified all-inclusive genre taxonomy have failed, researchers in MIR field tend to use simplified genre taxonomies typically including around 10 music genres.

Scaringella et al. [23] presented a survey on genre classification state-of-the-art and challenges. The authors reviewed the features of audio signal that researchers use for genre classification. These can be put into three classes that correspond to the main dimensions of music–timbre, melody/harmony, and rhythm.

- *Timbre* is defined as the perceptual feature of a musical note or sound that distinguishes different types of sound production, such as voices or musical instruments. The features related to timbre analyze spectral distribution of the signal. These features are low-level properties of the audio signal, and are commonly summarized by evaluating their distribution over larger temporal segments called texture windows, introduced by Tzanetakis and Cook [24].
- *Melody* is defined as the succession of pitched events perceived as single entity, and *harmony* is the use of pitch and chords. The features related to this dimension of music analyze pitch distribution of audio signal segments. Melody and harmony are described using mid-level audio features (e.g., chroma features) [25].
- *Rhythm* does not have a precise definition, and is identified with temporal regularity of a music piece. Rhythm information is extracted by analyzing beat periodicities of the signal.

Scaringella et al. [23] identified 3 possible approaches of implementing automatic genre classification—expert systems, unsupervised classification, and supervised classification.

- *Expert systems* are based on the idea of having a set of rules (defined by human experts), that given certain characteristics of a track assign it to a genre. Unfortunately, such approach is still not applicable to genre classification, since there is no fixed genre taxonomy and no defined characteristics of separate genres. Although there have been attempts to define the properties of music genres [22], no successful results have been achieved so far.
- *Unsupervised classification* approach is more realistic, as it does not require a fixed genre taxonomy. This approach is essentially a clustering method where the clusters are based on objective music-to-music similarity measures. These include Euclidean or Cosine distance between feature vectors, or building statistical models of feature distribution (e.g., using Gaussian Mixture Model), and comparing the models directly. The clustering algorithms typically used are: k-means, Self-Organizing Maps (SOM), and Growing Hierarchical Self-Organizing Maps (GHSOM) [26]. Major drawback of this approach is that resulting classification (or, more precisely, clustering) has no hierarchical structure and no actual genre labels.
- *Supervised classification* approach is the most used one, and relies on machine learning algorithms to map music tracks to a given genre taxonomy. Similarly to expert systems, the problem here is to have good genre taxonomy. The advantage of supervised learning, however, is that no rules are needed to assign a song to particular genre class—the algorithms learn these rules from training data. Most commonly used algorithms include the k-Nearest Neighbors (kNN), Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), and Support Vector Machine (SVM) classifiers [27].

The most significant contributions to genre classification research have been produced by techniques that used the supervised classification approach. Here we briefly present the milestone work by Tzanetakis and Cook [24], and a more recent work by Barbedo and Lopes [28].

Tzanetakis and Cook [24] set the standards for automatic audio classification into genres. Previous works in the area focused on music–speech discrimination. The authors proposed three feature sets representing the timbre, the rhythm, and the pitch properties of music. While timbre features were previously used for speech recognition, rhythm and pitch content features were specifically designed to represent the aspects of music—rhythm and harmony (melody). The authors used statistical feature classifiers (kNN and GMM) to classify music into 10 genres. The achieved accuracy was 61%.

Barbedo and Lopes [28] presented a novel approach to genre classification. They were the first to use a relatively low-dimensional feature space (12 features per audio segment), and a wide and deep musical genre taxonomy (4 levels with 29 genres on the lowest level). The authors designed a novel classification approach, where all possible pairs of genres were compared to each other, and this information was used to improve discrimination. The achieved precision was 61% for the lowest level of taxonomy (29 genres) and 87% on the highest level (3 genres).

In general, state-of-the-art approaches to genre classification cannot achieve precisions higher than 60% for large genre taxonomies. As current approaches do not scale to larger number of genre labels, some researchers look for alternative classification schemes. There has been work on classifying music into perceptual categories (tempo, mood, emotions, complexity, vocal content) [29]. Since such classification does not produce good results, researchers suggested the need for using extra-musical information like cultural aspects, listening habits, and lyrics to facilitate the classification task (see Section 2.4).

2.4. *Multimodal analysis in music information retrieval*

Multimedia data, and music in particular, comprises different types of information. In addition to the audio signal of music tracks, there are also lyrics, reviews, album covers, music videos, text surrounding the link to a music file. This additional information is rarely used in

[2] http://www.allmusic.com, http://www.amazon.com, http://www.mp3.com.

traditional MIR techniques. However, as MIR tasks are facing new challenges, researchers suggest that additional information can improve the performance of music retrieval or classification techniques. The research concerned with using other media types to retrieve the target media items is called *multimodal analysis*. An extensive overview of multimodal analysis techniques in MIR was given by Neumayer and Rauber [30].

Knopke [31] suggested using information about geographical location of audio resources to gather statistics about audio usage worldwide. Another work by the same author [32] described the process of collecting text data available on music web pages (anchor text, surrounding text, and filename), and analyzing it using traditional text similarity measures (TF-IDF, term weighting). The author argued that such information has a potential for improving music information retrieval performance since it creates a user-generated annotation that is not available in other MIR contexts. However, no actual implementation of this approach was presented in the work.

In a more recent work, Mayer and Neumayer [33] used the lyrics of songs to improve genre classification results. The lyrics were treated as a bag-of-words. The used features of lyrics include term occurrences, properties of the rhyming structure, distribution of parts of speech, and text statistic features (words per line, words per minute, etc.). The authors tested several dozens of feature combinations (both separately within the lyrics modality, as well as combined with audio features) with different classifiers (*k*NN, SVM, Naïve Bayes, etc.). The results showed the lyrics features alone to perform well, achieving classification accuracy similar to some of the audio features. Combining lyrics and audio features yielded a small increase in accuracy.

2.5. *Summary*

As pointed out by Scaringella et al. [23], extraction of high-level descriptors of audio signal is not yet state-of-the-art. Therefore, most MIR techniques are currently based on low-level signal features. Some researchers argue that low-level information may not be enough to bring music information retrieval closer to human perception of music, i.e., low-level audio features do not allow to capture certain aspects of music content [34,29]. This relates to the semantic gap problem, which is a core issue not only for music information retrieval, but for multimedia information retrieval in general [7].

Table 1 summarizes the tasks that traditional MIR techniques address. The most evolved areas of research are related to the usage of audio signal as a query. In such cases similarity search or classification can be performed by analyzing low-level features of music. However, there is a need for more high-level interaction with the user. The discussed MIR techniques cannot address such information needs of the users as finding a song by context information, emotional state, or semantic description. The new directions of MIR research that may help solving these tasks include contextual music retrieval and recommendation, affective computing, and social computing.

These new MIR directions are reviewed in detail in Section 4. Contextual recommendation and retrieval of music is a new research topic originating from the area of context-aware computing [35], which is focused on exploiting context information in order to provide the service most appropriate for the user's needs. We discuss this research area in Section 4.1.

Affective computing [36] is an area of computer science that deals with recognizing and processing human emotions. This research area is closely related to psychology and cognitive science. In music information retrieval affective computing can be used, e.g., to retrieve music that fits an emotional state of the user. Emotion recognition in music and its application to MIR is covered in Section 4.2.

Social computing is an area of computer science related to supporting social interaction between users. Furthermore, social computing exploits the content generated by users to provide services (e.g., collaborative filtering (see 3.1), tagging). We discuss social tagging application in music retrieval in Section 4.3.

3. Music recommendation

In this section we focus on music recommender systems. Music has been among the primary application domains for research on recommender systems. Attempts to recommend music have started as early as 1994 [37]—not much later than the field of recommender systems was born in the early 90's. The major breakthrough, however, came around at the turn of 2000's, when the World Wide Web became available to a large part of the population, and the digitalization of music content allowed major online recommender systems to emerge and create large user communities.

Music recommendation is a challenging task not only because of the complexity of music content, but also because human perception of music is still not thoroughly understood. It is a complex process that can be influenced by age, gender, personality traits, socio-economic, cultural background, and many other factors [38].

Similarly to recommender systems in other domains, music recommenders have used both collaborative filtering and content-based techniques. These approaches are sometimes combined to improve the quality of recommendations. In the following sections we will review the state-of-the-art in music recommender systems, and we will present the most popular applications implementing collaborative, content-based techniques, or a combination of the twos.

3.1. *Collaborative filtering*

Collaborative filtering (CF) is the most common approach not only for music recommendations, but also for other types of recommender systems. This technique relies on user-generated content (ratings or implicit feedback), and is based on the "word of mouth" approach to recommendations—items are recommended to a user if they were liked by similar users [37]. As a result, collaborative systems do not need to deal with the content, i.e., they do not base the decision whether to recommend an item or not on the description, or the physical properties of the item. In case of music recommendations it allows to avoid the task of analyzing and

Table 1 – An overview of traditional MIR tasks.

Information need	Input	Solution	Challenges
Retrieve the exact recording	Audio signal	Query by example	Unable to identify different recordings of the same song (e.g., cover versions); user may not be able to supply audio recording.
Retrieve a music track	Sung (hummed) melody	Query by humming	Only works for melodic music; user may be unable to supply good query; MIDI files of the recordings must be provided in the database.
Retrieve songs by genre, retrieve the genre of a song	Text query, audio signal	Genre classification	Precision not higher than 60%; no unified genre taxonomy.

classifying music content. This is an important advantage, given the complexity of the analysis of music signal and music metadata.

3.1.1. *General techniques*

The task of collaborative filtering is to predict the relevance of items to a user based on a database of user ratings. Collaborative filtering algorithms can be classified into two general categories—memory based and model based [39,40].

Memory based algorithms operate over the entire database to make predictions. Suppose U is the set of all users, and I the set of all items. Then the rating data is stored in a matrix R of dimensions $|U| \times |I|$, where each element $r_{u,i}$ in a row u is equal to the rating the user u gave to item i, or is null if the rating for this item is not known. The task of CF is to predict the null ratings. An unknown rating of user u for item i can be predicted either by finding a set of users similar to u (user-based CF), or a set of items similar to i (item-based CF), and then aggregating the ratings of similar users/items.

Here we give formulas for user-based CF. Given an active user u and an item i, the predicted rating for this item is:

$$\hat{r_{ui}} = r_u + K \sum_{v=1}^{n} w(u,v)(r_{vi} - r_v)$$

where r_u is the average rating of user u, n is the number of users in the database with known ratings for item $i, w(u,v)$ is the similarity of users u and v, K is a normalization factor such that the sum of $w(u,v)$ is 1 [39]. Different ways have been proposed to compute the user similarity score w [41]. The two most common are Pearson correlation (1) [42] and Cosine distance (2) [43] measures:

$$w(u,v) = \frac{\sum_{j=1}^{k} (r_{uj} - r_u)(r_{vj} - r_v)}{\sqrt{\sum_{j=1}^{k} (r_{uj} - r_u)^2 \sum_{j=1}^{k} (r_{vj} - r_v)^2}} \tag{1}$$

$$w(u,v) = \frac{\sum_{j=1}^{k} r_{uj} r_{vj}}{\sqrt{\sum_{j=1}^{k} r_{uj}^2 \sum_{j=1}^{k} r_{vj}^2}} \tag{2}$$

where k is the number of items both users u and v have rated.

Model based algorithms use the database of user ratings to learn a model which can be used for predicting unknown ratings. These algorithms take a probabilistic approach, and view the collaborative filtering task as computing the expected value of a user rating, given her ratings on other items. If user's ratings are integer values in the range $[0, m]$, the predicted rating of a user u for an item i is:

$$\hat{r_{ui}} = \sum_{j=0}^{m} \Pr(r_{ui} = j | r_{uk}, k \in R_u) j$$

where R_u is the set of ratings of the user u, and $\Pr(r_{u,i} = j | r_{u,k}, k \in R_u)$ is the probability that the active user u will give a rating j to the item i, given her previous ratings [39]. The most used techniques for estimating this probability are Bayesian Network and Clustering approaches [39,44].

In recent years, a new group of model-based techniques – known as matrix factorization models – has become popular in the recommender systems community [45,46]. These approaches are based on Singular Value Decomposition (SVD) techniques, used for identifying latent semantic factors in information retrieval.

Given the rating matrix R of dimensions $|U| \times |I|$, matrix factorization approach discovers f latent factors by finding two matrices – P (of dimension $|U| \times f$) and Q (of dimension $|I| \times f$) – such that their product approximates the matrix R:

$$R \approx P \times Q^T = \hat{R}.$$

Each row of P is a vector $p_u \in \mathbb{R}^f$. The elements of p_u show to what extent the user u has interest in the f factors. Similarly, each row of Q is a vector $q_i \in \mathbb{R}^f$ that shows how much item i possesses the f factors. The dot product of the user's and item's vectors then represents the user's u predicted rating for the item i:

$$\hat{r_{ui}} = p_u q_i^T.$$

The major challenge of matrix factorization approach is finding the matrices P and Q, i.e., learning the mapping of each item and user to their factor vectors p_u and q_i. In order to learn the factor vectors, the system minimizes the regularized squared error on the set of known ratings. The two most common approaches to do this are stochastic gradient descent [47] and alternating least squares [48] techniques.

Since memory-based algorithms compute predictions by performing an online scan of the user-item ratings matrix to identify neighbor users of the target one, they do not scale well for large real-world datasets. On the other hand, model-based algorithms use pre-computed models to make predictions. Therefore, most practical algorithms use either pure model-based techniques, or a mix of model- and memory-based approaches [44].

3.1.2. *Applications in the music domain*

In fact, some of the earliest research on collaborative filtering was done in the music domain. Back in 1994 Shardanand and

Please cite this article in press as: M. Kaminskas, F. Ricci, Contextual music information retrieval and recommendation: State of the art and challenges, Computer Science Review (2012), doi:10.1016/j.cosrev.2012.04.002

Maes [37] created Ringo—a system based on email message exchange between a user and the server. The users were asked to rate artists using a scale from 1 to 7, and received the list of recommended artists and albums based on the data of similar users. The authors evaluated 4 variations of user similarity computation, and found the constrained Pearson correlation (a variation where only ratings above or below a certain threshold contribute to the similarity) to perform best.

Hayes and Cunningham [49] were among the first to suggest using collaborative music recommendation for a music radio. They designed a client–server application that used streaming technology to play music. The users could build their radio programs and rate the tracks that were played. Based on these ratings, similar users were computed (using Pearson correlation). The target user was then recommended with tracks present in the programs of similar users. However, the authors did not provide any evaluation of their system.

Another online radio that used collaborative filtering [50] offered the same program for all listeners, but adjusted the repertoire to the current audience. The system allowed users to request songs, and transformed this information into user ratings for artists that perform these songs. Based on the user ratings, similar users were computed using the Mean Squared Difference algorithm [37]. Subsequently, the user–artist rating matrix was filled by predicting ratings for the artists unrated by the users. This information was used to determine the popular artists for current listeners.

Furthermore, the authors used item-based collaborative filtering [41] to determine artists that are similar to each other in order to keep the broadcasted playlist coherent. The artist similarity information was combined with popularity information to broadcast relevant songs. A small evaluation study with 10 users was conducted to check user satisfaction with the broadcasted playlists (5 songs per list). The study showed promising results, but the authors admitted that a bigger study is needed to draw significant conclusions.

Nowadays two of the most popular music recommender systems—Last.fm and Apple's Genius (available through iTunes[3]) exploit collaborative approach to recommend music content. We briefly review these systems in Section 3.4.

3.1.3. *Limitations*

CF is known to have problems that are related to the distribution of user ratings in the user-item matrix:

- "Cold start" is a problem of new items and new users. When a new item/user is added to the rating matrix, it has very few ratings, and therefore cannot be associated with other items/users;
- Data sparsity is another common problem of CF. When the number of users and items is large, it is common to have very low rating coverage, since a single user typically rates only a few items. As a result, predictions can be unreliable when based on neighbors whose similarity is estimated on a small number of co-rated items;
- The "long tail" problem (or popularity bias) is related to the diversity of recommendations provided by CF. Since it works on user ratings, popular items with many ratings tend to be recommended more frequently. Little known items are not recommended simply because few users rate them, and therefore these items do not appear in the profiles of the neighbor users.

In the attempts to solve these drawbacks of CF, researchers have typically introduced content-based techniques into their systems. We will discuss hybrid approaches in Section 3.3, therefore here we just briefly describe how the shortcomings of CF can be addressed.

Li et al. [51] suggested a collaborative music recommender system that, in addition to user ratings, uses basic audio features of the tracks to cluster similar items. The authors used a probabilistic model for the item-based filtering. Music tracks were clustered based on both ratings and content features (timbre, rhythm, and pitch features from [24]) using k-medoids clustering algorithm and Pearson correlation as the distance measure. Introducing the basic content features helped overcoming the "cold start" and data sparsity problems, since similar items could be detected even if they did not have any ratings in common. The evaluation of this approach showed a 17.9% improvement over standard memory-based Pearson correlation filtering, and a 6.4% improvement over standard item-based CF.

Konstas et al. [52] proposed using social networks to improve traditional collaborative recommendation techniques. The authors introduced a dataset based on the data from Last.fm social network, that describes a weighted social graph among users, tracks, and tags, thus representing not only users' musical preferences, but also the social relationships between the users and social tagging information. The authors used the Random Walk probabilistic model that can estimate similarity between two nodes in a graph. The obtained results were compared with a standard collaborative filtering approach applied to the same dataset. The results showed a statistically significant improvement over the standard CF method.

3.2. *Content-based approach*

While collaborative filtering was one of the first approaches used for recommending music, content-based (CB) recommendations in music domain have been used considerably less. The reason for this might be that content-based techniques require knowledge about the data, and music is notoriously difficult to describe and classify.

Content-based recommendation techniques are rooted in the field of information retrieval [53]. Therefore, content-based music recommenders typically exploit traditional music information retrieval techniques like acoustic fingerprint or genre detection (see Section 2).

3.2.1. *General techniques*

Content-based systems [54,53] store information describing the items, and retrieve items that are similar to those known to be liked by the user. Items are typically represented by n-dimensional feature vectors. The features describing

[3] http://www.apple.com/itunes/.

items can be collected automatically (e.g., using acoustic signal analysis in case of music tracks) or assigned to items manually (e.g., by domain experts).

The key step of content-based approach is learning the user model based on her preferences. This is a classification problem where the task is to learn a model, that given a new item would predict whether the user would be interested in the item. A number of learning algorithms can be used for this. A few examples are the Nearest Neighbor and the Relevance Feedback approaches.

The Nearest Neighbor algorithm simply stores all the training data, i.e., the items implicitly or explicitly evaluated by the user, in memory. In order to classify a new, unseen item, the algorithm compares it to all stored items using a similarity function (typically, Cosine or Euclidean distance between the feature vectors), and determines the nearest neighbor, or the k-nearest neighbors. The class label, or a numeric score for a previously unseen item can then be derived from the class labels of the nearest neighbors.

Relevance Feedback was introduced in information retrieval field by Rocchio [55]. It can be used for learning the user's profile vector. Initially, the profile vector is empty. It gets updated every time the user evaluates an item. After a sufficient number of iterations, the vector accurately represents the user's preferences.

$$q_m = \alpha q_0 + \left(\beta \frac{1}{|D_r|} \sum_{d_j \in D_r} d_j\right) - \left(\gamma \frac{1}{|D_{nr}|} \sum_{d_j \in D_{nr}} d_j\right)$$

here, q_m is the modified vector, q_0 is the original vector, D_r and D_{nr} are the set of relevant and non relevant items, and α, β, and γ are weights that are shifting the modified vector in a direction closer, or farther away from the original vector.

3.2.2. *Applications in the music domain*

Celma [56] presented FOAFing the Music—a system that uses information from the FOAF (Friend Of A Friend) project[4] to deliver music recommendations. The FOAF project provides conventions and a language to store the information a user says about herself in her homepage [57]. FOAF profiles include demographic and social information, and are based on RDF/XML vocabulary.

The system extracts music-related information from the interest property of a FOAF profile. Furthermore, the user's listening habits are extracted from her Last.fm profile. Based on this information, the system detects artists that the user likes. Artists similar to the ones liked by the user are found using a specially designed music ontology that describes genre, decade, nationality, and influences of artists, as well as key, key mode, tonality, and tempo of songs. Besides recommending relevant artists, the system uses a variety of RSS feeds to retrieve relevant information on upcoming concerts, new releases, podcast sessions, blog posts, and album reviews. The author, however, did not provide any system evaluation results.

Cano et al. [58] presented MusicSurfer—a content based system for navigating large music collections. The system retrieves similar artists for a given artist, and also has a query by example functionality (see Section 2.1). The authors argued that most content-based music similarity algorithms are based on low-level representations of music tracks, and therefore are not able to capture the relevant aspects of music that humans consider when rating musical pieces similar or dissimilar. As a solution the authors used perceptually and musically meaningful audio signal features (like rhythm, tonal strength, key note, key mode, timbre, and genre) that have been shown to be the most useful in music cognition research. The system achieved a precision of 24% for artist identification on a dataset with more than 11 K artists.

Hoashi et al. [59] combined a traditional MIR method with relevance feedback for content-based music recommendation. The authors used TreeQ [60]—a method that uses a tree structure to quantize the audio signal into a vector representation. Having obtained vector representations of audio tracks, Euclidean or Cosine distance can be used to compute similarity. The method has been shown to be effective for music information retrieval. However, large amounts of training data (100 songs or more) are required to generate the tree structure. The authors used TreeQ structure as a representation of user's preferences (i.e., a user profile). Since it is unlikely that a user would provide ratings for hundreds of songs to train the model, relevance feedback was used to adjust the model to user's preferences.

Sotiropoulos et al. [61] conjectured that different individuals assess music similarity via different audio features. The authors constructed 11 feature subsets from a set of 30 low-level audio features, and used these subsets in 11 different neural networks. Each neural network performs a similarity computation between two music tracks, and therefore can be used to retrieve the most similar music piece for a given track. Each of the neural networks was tested by 100 users. The results showed that, for each user there were neural networks approximating the music similarity perception of that particular individual consistently better than the remaining neural networks.

In a similar research Cataltepe and Altinel [62] presented a content-based music recommender system that adapts the set of audio features used for recommendations to each user individually, based on her listening history. This idea is based on the assumption that different users give more importance to different aspects of music. The authors clustered songs using different feature sets, then using Shannon entropy measure found the best clustering for a target user (i.e., the clustering approach that clusters the user's previously listened songs in the best way). Having determined the best clustering approach, the user's listening history was used to select clusters that contain songs previously listened by the user. The system then recommends songs from these clusters. Such adaptive usage of content features performs up to 60% better than standard approach with a static feature set.

3.2.3. *Limitations*

The limitations of content-based approaches are in fact those inherited from the information retrieval techniques that are reused and extended.

[4] http://www.foaf-project.org.

- The modeling of user's preferences is a major problem in CB systems. Content similarity cannot completely capture the preferences of a user. Such user modeling results in a semantic gap between the user's perception of music and the system's music representation;
- A related limitation is automatic feature extraction. In music information systems, extracting high level descriptors (e.g., genre or instrument information) is still a challenging task [23]. On the other hand, users are not able to define their needs in terms of low-level audio parameters (e.g., spectral shape features);
- The recommended tracks may lack novelty, and this occurs because the system tends to recommend items too similar to those that contributed to define the user's profile. This issue is somewhat similar to the "long tail" problem in CF systems—in both cases the users receive a limited number of recommendations that are either too obvious or too similar to each other. In the case of CF systems this happens due to the popularity bias, while in CB systems this occurs because the predictive model is overspecialized, having been trained on a limited number of music examples.

Content-based systems can overcome some of the limitations of CF. Popularity bias is not an issue in CB systems since all items are treated equally, independently of their popularity. Nevertheless, lack of novelty may still occur in CB systems (see above).

The "cold start" problem is only partly present in CB systems—new items do not cause problems, since they do not need to be rated by users in order to be retrieved by the system, however, new users are still an issue, since they need to rate sufficient number of items before their profiles are created.

3.3. *Hybrid approach*

As mentioned in the previous sections, the major problems of collaborative and content-based approaches are respectively new items/new users problem, and the problem of modeling user's preferences. Here we describe some research studies that combine collaborative and content-based approaches to take advantage, and to avoid the shortcomings of both techniques.

3.3.1. *General techniques*

An extensive overview of hybrid systems was given by Burke [63]. The author identified the following methods to combine different recommendation techniques:

- Weighted—the scores produced by different techniques are combined to produce a single recommendation. Let us say that two recommenders predict a user's rating for an item as 2 and 4. These scores can be combined, e.g., linearly, to produce a single prediction. Assigning equal weights to both systems would result in the final score for the item being 3. However, typically the scores are adjusted based on the user's feedback, or properties of the dataset;
- Switching—the system switches between the different techniques based on certain criteria, e.g., properties of the dataset, or the quality of produced recommendations;
- Mixed—recommendations produced by the different techniques are presented together, e.g., in a combined list, or side by side;
- Feature combination—item features from the different recommendation techniques (e.g., ratings and content features) are thrown together into a single recommendation algorithm;
- Cascade—the output of one recommendation technique is refined by another technique. For example, collaborative filtering might be used to produce a ranking of the items, and afterwards content-based filtering can be applied to break the ties;
- Feature augmentation—output of one recommendation technique is used as an input for another technique. For example, collaborative filtering may be used to find item features relevant for the target user, and this information later incorporated into content-based approach;
- Meta-level—the model learned by one recommender is used as an input for the other. Unlike the feature augmentation method, meta-level approach uses one system to produce a model (and not plain features) as input for the second system. For example, content-based system can be used to learn user models that can then be compared across users using a collaborative approach.

3.3.2. *Applications in the music domain*

Donaldson [64] presented a system that combines item-based collaborative filtering data with acoustic features using a feature combination hybridization. Song co-occurrence in playlists (from MyStrands dataset) was used to create a co-occurrence matrix which was then decomposed using eigenvalue estimation. This resulted in a song being described by a set of eigenvectors. On the content-based side, acoustic feature analysis was used to create a set of 30 feature vectors (timbre, rhythmic, and pitch features) describing each song. In total, each song in the dataset was described by 35 features and eigenvectors. The author suggested using weighted scheme to combine the different vectors when comparing two or more songs—feature vectors that are highly correlated and show a significant deviation from their means get larger weights, and therefore have more impact on the recommendation process.

The proposed system takes a user's playlist as a starting point for recommendations, and recommends songs that are similar to those present in the playlist (based on either co-occurrence, or acoustic similarity). The system can leverage social and cultural aspects of music, as well as the acoustic content analysis. It recommends more popular music if the supplied playlist contains popular tracks, co-occurring in other playlists, or it recommends more acoustically similar tracks if the seed playlist contains songs that have low co-occurrence rate in other playlists.

Yoshii et al. [65] presented another system based on feature combination approach. The system integrates both user rating data and content features. Ratings and content features are associated with a set of latent variables in a Bayesian network. This statistical model allows representing unobservable user preferences. The method proposed by the authors addresses both the problem of modeling the user's preferences, and the problem of new items in CF.

The evaluation results showed that the system provides more reliable and diverse recommendations compared to standard CF and CB approaches, and is able to recommend a small number of unrated tracks as well as the standard CB approach.

3.4. *Commercial music recommenders*

Commercial recommenders typically do not publish the details of the algorithms they use. Therefore, the following discussion is based on observations and on the information that can be found on the official web sites of the systems. It must be said that successful music recommenders most probably combine a variety of different techniques. However, certain systems are known to rely more on collaborative or on content-based approach. Table 2 reviews some of the popular systems.

3.4.1. *Collaborative-based systems*

Last.fm is a commercial Internet radio and recommender system that relies on collaborative filtering. The users of the system choose an artist as a starting point, and can then listen to a personalized radio station that recommends one song at a time. The system collects implicit feedback, i.e., the number of times a song was played, and explicit feedback, i.e., choosing a song as a favorite. This information is used to create a user profile that contains information about her listening habits. Last.fm finds similar users and then provides to the target user items that have been liked by similar users.

Overall, Last.fm gives good quality recommendations because of its large user community and user generated data. However, the system has problems recommending music that is not popular (i.e., belonging to the "long tail"), or music that is new and therefore not rated by a large number of users (i.e., the "cold start" problem).

Genius is a collaborative music recommendation software introduced as an additional feature in iTunes 8—a media player and a music organizer software from Apple Inc. Given a seed song, Genius generates a playlist of songs from the user's library that fit the selected song. Information about the user's library must first be sent to Apple's database. The system determines which songs to recommend based on the libraries of similar users, calculated from the set of libraries of iTunes users. The resulting Genius playlist can contain 25, 50, or 100 songs, and can be refreshed for obtaining new results, or saved.

Barrington et al. [66] analyzed the performance of Apple's Genius recommender. Although the authors did not have the access to details of the algorithm, they compared the output of the system – a generated playlist – with the outcomes of two other systems designed by the authors. The first system uses only artist similarity for generating playlists (using data from Last.fm), the second uses acoustic song similarity information. All systems use a single song as a starting point for generating a playlist. The evaluation experiment was conducted with real users, and the results showed that overall Genius produces the best recommendations, although for discovering the "long tail" (recommendations of less popular music) a content-based system can perform as well as Genius even without having the large amounts of data that is needed for collaborative filtering.

3.4.2. *Content-based systems*

Pandora[5] is the most popular content-based music recommender system. The user first provides a song or an artist's name, and then the system replies with recommendations. The recommendations are songs which have music qualities similar to the ones provided by the user. In order to determine the similarity, the system uses features of the songs from the Music Genome Project[6] database. The features are related to the sound properties of a track, for instance, minor key tonality, electric guitar riffs, instrumental arrangement, etc.—up to 500 attributes per track. Such detailed feature descriptions have been collected manually by music experts. The system may provide both successfully surprising recommendations, and unexpected songs (e.g., songs that are of different musical genre from the one liked by the target user). Users are allowed to provide feedback to the system. However, it is not clear how this feedback is used.

Musicovery[7] is an interactive Internet radio that uses audio content analysis to recommend music. Musicovery has two functionalities: the "mood pad"—a 2-dimensional continuous space that defines the mood of a song, and a graphical map of songs where tracks are displayed around the current song a user is listening to. The system can project any song on the "mood pad" space, where the axes represent arousal (Calm—Energetic) and mood (Dark—Positive) of a music track. Such idea of a 2-dimensional mood space is closely related to Thayer's mood model [67] (see Section 4.2.1). The user can navigate the system by clicking any point on the "mood pad", or by exploring the graphical map, where songs most likely to please the user appear closer to the current song. The systems allows users to ban songs, or mark them as favorites. However, it is not clear whether the feedback is used in the recommendation process.

3.5. *Summary*

To conclude this overview of the state-of-the-art in music recommender systems, it must be said that there is still an ongoing debate in the research community regarding the two major recommendation techniques—collaborative and content-based filtering. While it is generally accepted that the best solution is to mix the two approaches in hybrid systems, it is not clear which of the two techniques has more impact on good quality recommendations.

The question became even more important after the Netflix competition, where the winning algorithm did not use any content data. While these results have led some researchers to believe that content-based techniques have no future, most of people in music recommender research believe that lessons from Netflix do not necessarily apply to the music domain.

Some believe that collaborative techniques have limitations by design, therefore it is only a question of time when content-based techniques will evolve enough to take over the research and industry of music recommenders. Whether this

[5] http://www.pandora.com.

[6] http://www.pandora.com/mgp.shtml.

[7] http://musicovery.com/.

Table 2 – An overview of popular music recommenders.

System	Technique	User's input	Comments
Last.fm	Collaborative filtering	Favorite artist(s)	No audio analysis needed. Difficult to recommend music in the "long tail", "cold start" problem.
Genius	Collaborative filtering	A song	No audio analysis needed. Difficult to recommend music in the "long tail".
Pandora	Content-based	Favorite artist or song	No popularity bias, no "new item" problem. Content features assigned manually (scalability problem).
Musicovery	Content-based	Genre, a point on the 2-D mood space	Not clear if the system learns from user's feedback.

is true or not, we can admit that content-based music recommenders are still evolving, and collaborative techniques provide better results so far. However, in the future content-based approaches can play more important roles.

4. Contextual and social music retrieval and recommendation

In this section we focus on the newer trends of MIR research – context-aware computing, affective computing, and social computing – applied to the music domain. We first review research works that have exploited context information in music recommendation and retrieval, then we discuss the affective and social computing methods applied to MIR, since these methods are among the most used tools when solving modern music information retrieval problems (e.g., semantic, or context-aware music retrieval).

4.1. *Contextual music recommendation and retrieval*

Although not introduced in MIR research until recent years, the idea of using context information in computing applications has been introduced in the mid 90's. One of the first works in this area defined context as "*information describing where you are, who you are with, and what resources are nearby*" [35].

Dey [68] defined context in computing systems more formally as "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves*". In other words, context can be considered as any information that influences the interaction of the user with the system. For instance, in the domain of recommender systems context can be the situation of the user when searching for recommendations (e.g., time, mood, current activity). Clearly, such information may influence the information need of the user and thus it could be taken into account, in addition to the more conventional knowledge of the user's long term preferences, when providing recommendations [69].

Applications that can adapt or use context information to provide relevant information or services to the user are known as context-aware applications. Since being introduced by Schilit et al. [35], context-awareness became a popular research topic. Other synonyms for context-aware computing include adaptive, responsive, situated, context-sensitive, and environment-directed computing [68].

The idea of context-aware computing evolved from the ubiquitous (or pervasive) computing paradigm. The concept of ubiquitous computing was coined and defined by Weiser [70] as "*the method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user*". The idea of omnipresent computing accessible to the user at all times and in different conditions has created the basis for exploiting the context of both the user and the application.

Advanced technologies like smart devices and wearable computers created the opportunities for researchers to collect and use contextual (or situational) data to enrich the interaction between users and computers. Situational information like physical surroundings, time, emotional state, presence of other people, past and future events can help the system to better understand the current needs of the user.

The importance of context is even greater when users move away from traditional desktop computing environments. Thanks to mobile computing devices, users can access information and services in different surroundings and situations. The needs of users may vary depending on context, and context-aware systems can adapt to provide the relevant services.

Examples of context-aware applications include mobile computing systems (e.g., one of the first location-aware personal digital assistants, ParcTab [71]), adaptive user interfaces (e.g., an electronic white board for spontaneous meetings that records the data together with time and location information for future access [72]), augmented reality [73], tour guides (e.g., Cyberguide [74]), etc.

Although there has been research on using context-awareness in recommender systems [75], applying context-aware methods in the area of music recommendation is still relatively little researched. In this section we review the works that address contextual music recommendations. We structure the section based on the type of contextual information that has been used by the described systems.

There have been several attempts to provide the categorization of context information. Schilit et al. [35] divided context into: computing environment (available processors, network capacity), user environment (location, social situation), and physical environment (lighting and noise level). Dey [68,76] identified the four aspects of context that are most important: location, identity, activity, and time. These four dimensions are called the *primary context*, since they can be measured directly by various sensors and they serve as information sources for the higher-level context information, i.e., the *secondary context* (such as emotional state of the user). Razzaque et al. [77]gave an in depth discussion on context categorization and provided a hierarchical context classification that can be applied to any type of context-aware applications.

Since our review focuses on a specific domain (music), we adopt a simpler classification of contextual information: environment-related context (information about the location of the user, the current time, weather, temperature, etc.), user-related context (information about the activity of the user, the user's demographic information, emotional state), and multimedia context (other types of information the user is exposed to besides music, e.g., text, images). Table 3 gives an overview of the systems that exploit these different types of contextual information.

4.1.1. *Environment-related context*

In this section we review research that exploits the knowledge of users' environment parameters like season, temperature, time, weather conditions, etc. in the music recommendation domain. The motivation for such research is based on the fact that the environment has an influence on the state-of-mind or emotional state of the user, and therefore may indirectly influence the musical preferences. Research has shown that there exists a positive relation between the affective qualities of the listening situation and the preference for music that augments these qualities [88]. For instance, people tend to prefer different types of music in summer and in winter [89]. Consequently, environment-related parameters have to be considered when recommending music content.

The environment-related context used in music recommendation research can be classified into the following general categories:

- *Location* of the user. It can be represented by a ZIP code, geographical coordinates, type of landscape (e.g., city, nature), nearby monuments, buildings, landmarks, etc. The surroundings of the user may have a strong impact on her perception and preferences of music. For instance, walking down a busy city street evokes different emotions (and consequently music preferences) compared to a stroll in the woods. The US music duo Bluebrain is the first band to record a location-aware album.[8] In 2011 the band released two such albums—one dedicated to Washington's park National Mall, and the second dedicated to New York's Central Park. Both albums were released as iPhone apps, with music tracks pre-recorded for specific zones in the parks. As the listener moves through the landscape, the tracks change through smooth transitions, providing a soundtrack to the walk. Despite the large potential of location-aware music services, up to date there has been little research exploring location-related context information in music recommendations.
- *Time*-related information can refer to the time of day (typically categorized into morning, afternoon, evening, night), or day of week (can be represented by the exact day or can be categorized into working day, weekend). Such context information is potentially useful since the user's music preferences may be different on a busy Monday morning compared to a relaxed Saturday afternoon.
- *Weather*-related information. It may refer to weather conditions (typically categorized into sunny, overcast, rainy, etc.), to the temperature (e.g., cold, moderate, hot), or to the season. Such information is relevant for music recommendations since the user's music preferences may significantly differ, e.g., in a cold rainy autumn or a hot sunny midsummer [89].
- Other parameters, such as *traffic*, *noise* level, or *light* level contribute to the emotional state of the user and therefore indirectly influence her music preferences.

Reddy and Mascia [78] presented a mobile music recommender system Lifetrak that generates a playlist using the user's music library based on the current context of the user. The environment-related context information used by the authors includes location (represented by a ZIP code), time of day (morning, afternoon, evening, night), day of week, noise or traffic level – depending on whether the user is walking or driving – (calm, moderate, chaotic), temperature (frigid, cold, moderate, warm, hot), and weather (rain, snow, haze, cloudy, sunny, clear). The context is obtained using the sensors of the mobile device that stores the application (the GPS for location and speed, the clock for the time, the microphone for the noise level) and RSS feeds (for weather and traffic conditions).

In order to generate recommendations, the songs in the user's library have to be tagged by the user with tags from a controlled vocabulary. The tags represent the values of the previously mentioned context parameters. Given the library of tagged songs and the current context of the user determined using the sensors, the system ranks all the songs in the library based on their match with the context. The importance of each context dimension has a default value of 1 and can be altered either explicitly by the user, or exploiting the user's feedback—if the user expresses love or hate for a song in a certain context, the importance of all the current context dimensions for that song are increased or decreased by 0.1.

Unfortunately, the above mentioned approach was not evaluated. Consequently, it remains unclear whether the system produces good quality recommendations, whether the context features have been chosen appropriately, and if the values in the user's feedback computation were tuned. Extensive real-world testing remains to be performed to answer these questions. Another limitation is related to the labeling of music tracks with appropriate tags. This task requires a lot of user's effort and therefore might discourage some users from using the system.

Park et al. [79] presented a context-aware music recommender where the environment-related context information includes temperature, humidity, noise, light level, weather, season, and time of day. The authors used Bayesian networks to infer the emotional state of the user based on contextual data. The emotional state of the user is assigned to one of the following classes: Depressing, Content, Exuberant, or Anxious/Frantic.

The music tracks used in the described system are represented by genre (5 genres), tempo (3 attributes), and mood (9 attributes)—17 attributes in total. In order to relate music to the emotional state classes, users must explicitly express their preferences for each music attribute in all

[8] http://bluebrainmusic.blogspot.com/.

Table 3 – An overview of context-aware music recommenders.

System	Context features	Implementation and evaluation	Limitations
Lifetrak [78]	Location, time of day, day of week, noise and traffic level, temperature, weather, activity of the user.	Prototype implemented for Nokia 770 Internet Tablet.	No evaluation, high user effort to input information.
CA-MRS Using Fuzzy Bayesian Networks and Utility Theory [79]	Temperature, humidity, noise, luminance, weather, season, time of day, user's gender, and age.	Prototype implemented for a desktop PC. An evaluation study performed with 10 users.	Limited evaluation, high user effort to input information.
Sonic City [80]	Light, noise, and pollution level, temperature, electromagnetic activity, enclosure, slope, presence of metal, heart rate, pace, ascension/descent.	A wearable prototype implemented. An evaluation study performed with 5 users.	Difficult to use in everyday settings. Only specific genre of music supported.
Music for my mood [81]	Season, month, weekday, weather, temperature, gender, age, region.	Desktop PC implementation. Offline evaluation performed with 660 users.	The system only distinguishes between slow and fast music. Only four music genres supported.
Adapting music to POIs [82,5]	Places of interest.	A desktop simulation study with 10 users, followed by a real-world study with 26 users testing an Android travel guide.	The matching of music and POIs is based on manually assigned tags. Therefore, the approach needs to be expanded to perform the matching automatically.
Foxtrot [83]	Location.	Desktop prototype implemented. Desktop simulation performed with 100 users.	The system was not compared to an alternative approach. User community is essential for obtaining the data.
Personal Soundtrack [84]	The user's walking pace.	A prototype implemented for a laptop PC with accelerometer connected via Bluetooth. Evaluation process described, but no results given.	The prototype is difficult to use since one needs to carry the laptop. No evaluation results.
MusicSense [85]	Text of the viewed web page.	Evaluation performed with 100 songs and 50 Weblogs.	Not all web documents are emotion driven. Music represented only by text (lyrics and reviews).
Emotion-based impressionism slideshow [86]	Images.	Evaluation performed with 138 impressionism paintings and 160 classical piano compositions.	Limited evaluation.
Picasso [87]	Images.	Evaluation performed with 13 users.	The training dataset of movie screenshots and soundtrack pieces is expensive to obtain.

possible contextual dimensions. For example, a user may state that she prefers rock genre with the utility weight of 4 in a depressing state, weight 3 in a content state, weight 2 in an exuberant state, and weight 3 in an anxious/frantic state.

Given this information and the values of current context parameters, the system can infer the current emotional state of the user, compute the score of each music track for the current state, and recommend the top-N music tracks. For evaluation purposes, the system was compared with random recommendations. A study with 10 users showed that the users were more satisfied with the tracks recommended by the context-aware system. Similarly to the work of Reddy and Mascia [78], a major limitation of the system is related to high user effort required to state the music preferences.

An attempt to combine environmental-related and user-related context information for interactive music generation was presented by Gaye et al. [80]. The described system takes a wide range of both environment-related and user-related input parameters for generating electronic music in real-time. The authors defined the concept of *mobility* as the union of environment parameters and user's actions. The used environment parameters are: light level, noise level, pollution level, temperature, electromagnetic activity, enclosure, slope, and presence of metal. These input parameters can be discrete (e.g., a car passing) or continuous (e.g., light level).

In order to map the contextual parameters to music features, the authors used two layers of abstraction of the input parameters—high-level and low-level input. The high-level input influences the composition structure of the generated music, and the low-level input influences either short events in the music track (in the case of discrete input), or spectral music parameters (in the case of continuous input). The authors have implemented a wearable prototype that consists of sensors, a micro controller, and a laptop with a music programming environment. A small evaluation study was conducted [90] with 5 users trying the prototype in everyday settings. The main goal of the study was to check whether the users perceive the device as a dynamic walkman, a music instrument, or an exploration tool.

The results showed that although the device enhanced the users' perception of surroundings, the users also wanted more control over the music generation process. The major drawbacks of the system are hardware-related—a complex network of sensors and wires makes it difficult to use in everyday life. Furthermore, due to the specific genre of artificially generated music (electronic), this device is not suitable for a wide range of music listeners.

Lee and Lee [81] used the environment-related context data to improve the performance of a music recommender system. The authors used a case-based reasoning approach assuming that if a user listened to certain music in certain contextual conditions, she will want to listen to the same (or similar) music in the same (or similar) context conditions. Therefore, the described system determines the user's intention to listen to music and the type of preferred music by analyzing previous cases with similar contextual conditions.

The authors used a listening history dataset of an online music streaming service. The values of context parameters – season, month, weekday, weather, and temperature – were obtained from a weather bureau. Given a user and certain contextual conditions, the system finds *k* most similar previous cases in the user's listening history and suggests the music that the active user listened to in those cases. Following the first publication, the authors have extended their system to suggest music not only listened by the same user in similar contextual conditions, but also the music listened by similar users in similar conditions [91]. The user-to-user similarity is based on the users' demographical data (gender, age, region).

The evaluation of the proposed approach was performed by comparing the system with a case-based reasoning music recommender that does not use context information. An offline evaluation showed an improvement of 8% in the average precision when using contextual data. A limitation of the described system is a shallow music classification with only four genres and two classes—slow (ballad, R&B) and fast (rock/metal, dance). In other words, the system can only distinguish between slow and fast music. Consequently, the produced recommendations may be unsatisfying to users with specific music preferences.

Kaminskas and Ricci [82] explored the possibilities to adapt music to the places of interest (POIs) that the user is visiting. This idea is based on the hypothesis that a fitting music track may enhance the sightseeing experience of the user. For instance, during a visit to a Baroque cathedral a user might enjoy hearing a composition by Bach, while the narrow streets in Venice offer a good surrounding to listen to Vivaldi's concerto.

The matching of music and POIs was made by representing both music tracks and POIs with a common set of tags, and comparing the tag profiles of the items. The used vocabulary of tags largely consisted of emotion labels taken from a music perception research [92]. The authors claim that both music and places can raise emotions, and the commonality of the raised emotions could provide the base for establishing a degree of match between a place and a music track.

In order to test the hypothesis that users agree with the established match between a music track and a POI, the authors have performed an experiment where subjects were repeatedly presented with POIs and a selection of music tracks, some of them matching the presented POI, and some not. The users were requested to mark those tracks that they judged to be suited for the illustrated POI. The results of this experiment showed that there is a strong overlap between the users' selections and the best matching music that is recommended by the system for a POI. Following the initial experiment, a mobile travel guide was implemented and evaluated in a real-world scenario where the system suggested an itinerary and played recommended music for each visited POI [5]. The results of the study showed that users judge the recommended music suited for the POIs. Moreover, the music was rated higher when it was played in this usage scenario.

Ankolekar and Sandholm [83] presented a mobile audio application, Foxtrot, that allows its users to explicitly assign audio content to a particular location. Similarly to Kaminskas and Ricci [82], this work also stressed the importance of the emotional link between music and location. According to the authors, the primary goal of their system is to enhance the sense of being in a place by creating its emotional atmosphere. However, instead of using a knowledge-driven approach, i.e., understanding which emotional characteristics link music and a location, Foxtrot relies on crowd-sourcing—the users of Foxtrot are allowed to assign audio piece (either a music track or a sound clip) to a specific location (represented by the geographical coordinates of the user's current location), and also specify the visibility range of the audio track.

The system is then able to provide a stream of location-aware audio content to the users in the following way: given a user's location, speed and trajectory, the system selects tracks in close geographic proximity (if the user is moving, tracks closer to her expected trajectory are preferred over those close to the initial location), and ranks them according to a relevance score. The score is computed as a weighted combination of its likability (estimated from the user's explicit feedback and listening history of the track), geographic relevance (determined by the proximity and visibility of the track), and freshness (estimated from the user's listening history).

For evaluation purposes, a desktop version of Foxtrot has been implemented for the city of Palo Alto, CA, USA. A set of 30 s music clips as well as specially recorded ambient sounds were used for the evaluation. The evaluation of the system was performed with 100 users by means of the Mechanical Turk environment.[9] The users were asked to complete a 3-min virtual walk in Palo Alto. Three alternative versions of the system were used in the study – the first one suggesting only ambient sounds, the second suggesting only music tracks, and the third – a mixture of both sounds and music. Results of the study showed that users were more engaged with the system and reported a better emotional experience when listening to music clips only. On the other hand, ambient sounds were reported to provide a better walking experience, i.e., creating a feeling of actually being at the location. However, we note that the system was not compared to an alternative baseline not exploiting contextual information, for instance a standard radio station of the area.

[9] https://www.mturk.com:443/mturk/welcome.

4.1.2. *User-related context*

The user-related context used in music recommendation research can be classified into the following categories:

- *Activity* (or the state) of the user. Such information may include an action which is typically represented as an element from the set of possible actions (e.g., walking, running, driving), or a numerical parameter defining the user's state (e.g., walking pace or heart rate). This type of context may have a direct impact on the user's musical preferences—the same person might prefer different music when relaxing compared to when working out in a gym. For instance, Foley [93] has shown that people preferred different musical tempo depending on their occupation.
- *Demographical* information about the user. Although this type of information is sometimes viewed as part of the user's profile, it can be considered as a user-related context. It may include the user's age, gender, nationality, social group, personality traits, etc. In the field of music psychology there have been studies that related personality traits and social lifestyles to music preferences [94].
- *Emotional state* of the user. Our mood has a direct influence on our musical preferences. For example, we all listen to very different music in a sad mood compared to when being happy. Although some earlier research on music psychology suggested that people select music that moderates their emotional condition [95], newer works suggest that music is selected so as to augment the emotions perceived by the listener [88]. Since music is an emotion-loaded type of content often described by emotions, there is a direct match between the user's mood and the preferred music.

User-related context is typically used in combination with the environment-related context. For example, most works mentioned in the previous section use activity context and demographical context in combination with the environment-related context. Reddy and Mascia [78] used the activity of the user (walking, running, driving), Park et al. [79] used the user's gender and age, Gaye et al. [80] used heart rate, pace, ascension, descent.

Baltrunas et al. [6] focused on a specific music adaptation scenario—recommending music while driving a car. The authors identified the following contextual conditions that might have influence on the user's music preferences in this setting: driving style, road type, landscape type, sleepiness level, traffic conditions, mood, weather, and time of day.

The authors focused on implementing context-awareness in a collaborative filtering recommender (see Section 3.1)—a technique based on assigning to each rating the contextual conditions that the rating was acquired in [75]. The main contribution of this work is the description of a methodology for supporting the development cycle of a context-aware collaborative filtering recommender system. The methodology comprises four steps: context factors relevance assessment, in-context acquisition of ratings, context-aware rating prediction, and context-aware recommendation generation.

The ratings have been acquired using a web application where the users were asked to rate 139 music tracks (belonging to 10 genres) without assuming any particular context, and also imagining different contextual conditions. Following the acquisition of in-context ratings, a prediction model extending matrix factorization approach [45] has been trained. An offline evaluation of the model has shown an improvement of 3% over standard matrix factorization prediction method. A prototype system was developed as a mobile Android application, which allows the users to set the values of the contextual conditions manually, listen to the recommended tracks, and rate them. The live evaluation of the system is left for future work.

Elliott and Tomlinson [84] presented a research that focuses strictly on the user's activity context. The authors designed a system that adapts music to the user's walking pace by matching beats-per-minute (BPM) of music tracks with the user's steps-per-minute. The system uses implicit user's feedback by estimating the likelihood of a song being played based on the number of times the user has previously skipped this song. In this first prototype the BPM was manually calculated for songs from the user's library. The system was implemented on a laptop device with accelerometer connected via Bluetooth. The authors described an ongoing evaluation where they were testing a hypothesis that matching the person's pace with the tempo of a song can provide greater satisfaction for the listener compared to a randomly generated playlist. However, the results were not presented in the paper.

The relation between the user's activity and music can also be researched in the opposite direction: instead of using the user's activity to infer the musical preferences, music can be used as an aid in the activity. For instance, Jones et al. [96] exploited the practice of listening to music while being mobile to provide navigation cues. The proposed system adapts volume and spatial balance of played music to lead users to their destinations. The authors described a prototype running on a pocket PC attached to a GPS device. A small field study with 10 users showed a 86% success rate in completing navigation tasks. However, the device was not compared to other navigation support techniques in this field study. Despite not being directly related to music recommendation research, this work shows the potential of using music in location-aware services.

Emotional state of the user is an example of a *secondary context* [76], since it cannot be measured directly, but needs to be derived from other types of contextual information. Consequently, mood is often not modeled as a separate context feature, but rather used as an abstract layer for classifying the various factors (environment- or activity-related) that influence the user's emotional state. For instance, Park et al. [79] used the different emotional states as classes into which both music and the contextual conditions can be classified. Also, Kaminskas and Ricci [82,5] used emotions when matching music to places of interest.

In this way, emotions act as a bridge between music and items belonging to another domain. In fact, almost any type of content that can be linked to, or tagged with a mood, can also be matched with music through the mood. We already saw how environment factors can be classified into emotion categories. In the next section, we will see that also multimedia content – text and images – can be matched with music via emotions. A more detailed discussion on emotions and music is given in Section 4.2.

4.1.3. *Multimedia context*

Since music is a type of multimedia content, it is often useful to combine it with other types of multimedia. Such combination can be used to enhance information presentation, cross-selling of entertainment items, etc. In music adaptation research, music has been adapted to the following types of multimedia content:

- *Text* is the simplest form of multimedia information. It can be related to music considering its semantic or emotional content. For instance, both written text and a song may talk about the same historical event, or both may convey the same emotions. Moreover, songs have lyrics and reviews that are represented as text.
- *Images* are a more complex form of information. Like text, images can be related to music on semantic and emotional levels. This is especially true for paintings as they typically carry emotional messages.

Cai et al. [85] worked on adapting music to text content. The authors obtained some positive results with a system that recommends music to users while they are reading web documents (particularly weblogs). The idea (inspired by Google's AdSense) is to automatically propose music which is relevant to the web page the user is reading. In such scenario, the text of the web page is the context in which music is recommended.

In order to match music and the content of a web page, both resources are represented as text documents (using the bag of words model). The text describing music is obtained by parsing the lyrics and online reviews. Both documents are then mapped to a common emotion space. The authors presented a generative model for determining the emotions of a text document. Given the emotion distributions of the two documents – one representing a music track and the other representing a web page –, their similarity is computed by comparing the two distributions. For a web document, the top-N most similar music tracks are recommended.

The evaluation of the proposed approach consisted of two parts—evaluating the performance of emotion modeling, and evaluating the music recommendation quality. For the first part, the ground truth was obtained by manually labeling a corpus of 100 songs and 50 weblogs with emotions from a controlled vocabulary. For the second part, the ground truth was obtained by manually assigning 3–5 songs (from the set of 100 songs) to each of the 50 weblogs. The results of emotion modeling were compared with the manually labeled songs and weblogs. The results showed an average correlation of 0.48 for songs and 0.42 for weblogs. The results of music recommendation evaluation showed the precision of 42% and recall of 72% for the top-10 recommendations. The limitations of the proposed approach are the following: not all web documents are clearly connected with emotions (although weblogs often are). Furthermore, representing music only by text is limited (although the authors claim that their approach can be expanded to take audio features as parameters).

Li and Shan [86] presented a research on adapting music to images. The authors proposed a way to generate an automatic music accompaniment for a slideshow of impressionism paintings. Similarly to the previous research, the matching of music tracks and context (in this case paintings) is based on the affective properties of the objects. The emotions of paintings are determined by discovering the associations between emotions and features of paintings from the training data – paintings manually labeled with emotions – using the Mixed Media Graph (MMG) model, originally used for finding correlations between multimedia objects [97]. The used painting features are related to color, light, and texture properties of the image. Similarly, the associations between music track features and emotions are discovered by training the MMG on a manually labeled training set of tracks. The used music features are related to melody, rhythm, and tempo.

The authors did not describe the emotion labels they had assigned to paintings and music. However, it is said that the labels were derived from Russell's circumplex emotion model [98]. Being able to assign emotion labels to both paintings and music, the adaptation process is as follows. First, the features of paintings are extracted and the emotions determined. Then, the paintings are clustered based on their emotions. Finally, the emotions of music tracks are determined, and, for every cluster of paintings, a single music track is selected.

The used dataset consists of 138 impressionism paintings and 160 classical piano compositions. The authors did not provide a clear description of the evaluation procedure. They seem to have compared a slideshow generated by their approach with slideshows produced by ACDSee and Photo Story software. However, it is not clear how music was assigned to the software-generated slideshows. 18 users were asked to view different slideshows and evaluate the audiovisual satisfaction. The results showed that the proposed approach provides a greater satisfaction to the users.

More recently, Stupar and Michel [87] presented PICASSO—a system that retrieves music given a query image by exploiting movie soundtracks. The authors argued that in movies the soundtrack music is adapted to the shown imagery, and this information can be used for matching music tracks with images. From a selection of 28 movies, the authors have created a database of over 40,000 movie screenshots, each of them associated with a musical piece from the movie soundtrack. The proposed approach to recommend music for an image is as follows: given the query image, the system finds the most similar movies' screenshots, retrieves the soundtrack parts played while these screenshots were taken, and suggests music tracks (e.g., from the user's music collection) similar to these retrieved soundtrack pieces.

In order to find screenshot images similar to the query image, the authors computed image-to-image similarity based on MPEG-7 low-level color and texture features. For finding music tracks most similar to the soundtrack pieces, music-to-music similarity was computed using low-level spectral audio features (MFCCs, spectral centroid, spectral rolloff) and chroma features [25].

For evaluating the approach the authors have conducted a study with 13 users, where each user was asked to rate the appropriateness of the first ranked, the tenth ranked, and a randomly suggested music track for a given image. A set of 12 images was used in the study, and music was selected from a collection of 275 tracks. The results showed that users rate higher the appropriateness of tracks matched to the image compared to the randomly selected tracks. The authors have also performed an evaluation of soundtrack generation for a sequence of images, which produced similar results.

4.1.4. *Summary*

In conclusion, we must observe that most of the approaches that relate music to contextual conditions are data-driven. In other words, the authors tend to use a mix of contextual parameters in a machine learning algorithm without trying to understand the relations between music and certain contextual conditions.

An alternative would be to consider knowledge-driven approaches, where music selection is influenced by certain contextual factors and by the knowledge about the relationships between context and music, e.g., how the contextual conditions influence music perception.

More research is needed to understand what relates certain contextual conditions to music or its individual features. Although this topic has been researched in the field of music psychology [93,95,89,94,88], collaboration is needed between music psychologists and researchers in the area of music recommendation. The knowledge derived from such collaboration could help not only to improve the quality of contextual music recommendations, but also to extend the application area of this research.

4.2. *Emotion recognition in music*

Music is generally considered an emotion-oriented form of content—it conveys certain emotions to the listener, and therefore can be characterized by emotion labels. For instance, most people will agree with labeling Rossini's "William Tell Overture" as *happy* and *energetic*, and Stravinsky's "Firebird" as *calm* and *anxious* [99]. Being able to automatically detect the emotions conveyed by any given music track is a challenging task that has received considerable attention in the music information retrieval community [100]. Moreover, since it deals with human perception of music and emotion modeling, the topic of automatic emotion recognition is closely related to the field of music psychology.

Automatic emotion recognition in music can be applied both to music retrieval and recommendation. For instance, users may search for music that conveys particular emotions, or they may want to receive music recommendations based on their current mood.

Kim et al. [100] gave an extensive review of the state-of-the-art in music emotion recognition. It covers the topics of psychology research on emotion, collecting emotion annotations for music content, content-based audio analysis, and multimodal analysis for emotion recognition. In this section, we give a briefer overview of the research area. We first present psychology-based research on emotion models, and then review some of the research works that have used machine learning to detect emotions in music content.

4.2.1. *Emotion models for music cognition*

The definition of the possible emotions is a prerequisite for automatically detecting emotions conveyed by a music track. However, similarly to music genres, the set of different emotions conveyed by music is not easy to determine. Despite numerous researches in the domain of cognitive psychology, up to date no universal taxonomy of emotions has been agreed on. Human emotions are complex and multi-layered, therefore, focusing on different aspects of emotions leads to producing different lists of emotions. This means that there may not exist a universal emotion model to discover, but emotions are to be chosen based on the task and domain of the research [101].

Dunker et al. [102] reviewed emotion models that can be applied to music and images. The authors distinguished two groups of models: *category-based models*, where sets of emotion terms are arranged into emotion categories, and *dimensional models*, where emotions are represented as a combination of different dimensions. Here we present some of the better known models of both types.

Category-based models. One of the earliest emotion models was presented by Hevner [103]. The author has researched the relation between certain emotions and the properties of music, e.g., mode, rhythm, harmony, etc. During a user study the subjects had been given a list of 67 emotion adjectives to tag classical music compositions. The author later arranged the adjectives into 8 clusters with the representative emotions of the clusters being: *dignified*, *sad*, *dreamy*, *serene*, *graceful*, *happy*, *exciting*, and *vigorous*.

The adjective list introduced by Hevner has been extensively used as a reference set for measuring emotional response to music. The list was later revised and expanded by Farnsworth [104] who attempted to improve the consistency of emotions inside each cluster. Based on the results of a user study, where 200 users tagged 56 musical pieces using the list of adjectives, the emotions were re-arranged into 9 clusters, leaving out 16 of the original 67 adjectives. Here we present an adjective from each of the nine clusters: *happy*, *light*, *graceful*, *dreamy*, *longing*, *sad*, *spiritual*, *triumphant*, and *vigorous*. Subsequently, the author added a tenth cluster with only one emotion in it—*frustrated*.

Further revision of the list was presented by Schubert [105]. The author produced a list of emotion adjectives consisting of the 67 words originally used by Hevner and 24 additional words taken from non-musical emotion models—Russel's circumplex model [98], and Whissel's dictionary of affect [106]. The resulting list of 91 adjectives was evaluated by 133 musically experienced users. Unlike previous user studies, in this study the users were not asked to tag certain music tracks, but rather evaluate the appropriateness of each adjective for describing music (i.e., to imagine if a music described by such word exists). Thus, the study avoided a bias towards a selection of music tracks. The outcome of this study was a list of 46 emotion adjectives arranged into 9 clusters. Compared to the previous models of Hevner and Farnsworth, some of the words dropped by Farnsworth were reintroduced, and other 15 words (among others *frustrated*, *martial*, *pleased*) were dropped as not suitable to describe music. A list of representative adjectives from each of the nine clusters follows: *happy*, *light*, *graceful*, *dreamy*, *sad*, *vigorous*, *tragic*, *angry*, and *triumphant*.

A more recent and elaborated research on emotional response to music was carried out by Zentner et al. [92]. Four large scale user studies were performed to determine which emotions are perceived and felt by music listeners. The goal of the first study was to establish a list of adjectives that are potentially useful in describing emotions. 92 psychology students have evaluated 515 candidate terms by answering the question "*does this adjective describe an internal affective state?*" using a binary yes/no scale. The outcome of this study was a list of 146 affect terms.

During the second study that list of 146 adjectives was evaluated by 262 participants. The participants rated the adjectives by measuring how often they feel and perceive the corresponding emotions when listening to music. After eliminating the adjectives that were rated low by the users, a list of 89 terms was obtained. The third and fourth studies were conducted to further reduce the list of emotions perceived by the listeners and to cluster the emotion terms using factor analysis. The final set of 33 emotion terms in 9 clusters was called the Geneva Emotional Music Scale (GEMS) model. The representative emotions of the clusters are: *wonder*, *transcendence*, *tenderness*, *nostalgia*, *peacefulness*, *power*, *joyful activation*, *tension*, and *sadness*.

A limitation of this study is related to the language used during the research. The four user studies have been conducted in French. Consequently, inconsistencies might have been introduced when translating the results into English. Nevertheless, up to date this study remains the most extensive real user study on emotions evoked by music.

Dimensional models. Dimensional models are an alternative to the category-based emotion models that allows to represent emotional states as a combination of a small number of independent dimensions. Although there have been attempts to use three dimensions [36], the most common models include two dimensions. Researchers gave different names to these dimensions, however, they all tend to agree on a common idea of modeling emotional states as a combination of activeness and positiveness of emotion. Thus, the first dimension represents the *Activation* level (also called *Activity*, *Arousal* or *Energy*), which contains values between *quiet* and *energetic*; and the second dimension represents the *Valence* level (also called *Stress* or *Pleasure*), which contains values between *negative* and *positive*.

The famous 2-dimensional emotion models are: Russell's circumplex model [98], Thayer's model [67], and Tellegen–Watson–Clark (TWC) model [107]. Fig. 1 shows an approximate matching of these three models.

Russell [98] designed a model where emotions are placed in a circle on a 2-dimensional space. The dimensions are called *Arousal* and *Pleasure*. Later, Thayer [67] modeled emotional states using the *Energy* and *Stress* dimensions. Compared to these two models, in TWC model [107] the axes of the two dimensions are turned at a 45° angle in relation to the *Valence* and *Arousal* axes. This way, one dimension combines the positive valence with high arousal and is called *Positive Affect*, while the other combines negative valence with high arousal and is called *Negative Affect*.

Other researchers have based their work on the idea of 2-dimensional emotion space. For example, Whissell [106] defined a set of emotion labels that can be described using real values in 2 dimensions (e.g., *calm* has the activation value of 2.5 and the valence value of 5.5, while *anxious* has the activation of 6 and the valence of 2.3).

Plutchik [108] presented another 2D model where emotions are not evenly distributed in the 2-dimensional space, but rather arranged in a circle. Thus for each emotion label, only an angle is required (e.g., *accepting* being at 0°, and *sad* being at 108.5°).

Despite being a useful abstraction, 2-dimensional models have limitations—collapsing such complex domain as human emotions into two dimensions leads to information loss. This is illustrated by the fact that *fear* and *anger* appear at the opposite sides of Plutchik's emotion circle, but are close to each other in Whissell's emotion space. In summary, both category-based and dimensional approaches to emotion modeling have their strong and weak points. Dunker et al. [102] concluded that category-based models are suitable for applications that rely on tagging (see Section 4.3), and, moreover, provide the hierarchical vocabulary structure (i.e., emotion terms grouped into clusters) which is useful when varying the level of details in emotion analysis. On the other hand, dimensional models are better for the visualization of emotions, and can be very useful in visual navigation and content discovery systems (e.g., Musicovery, described in Section 3.4).

4.2.2. *Machine learning approaches to emotion recognition in music*

Being able to automatically detect the emotions conveyed by a music track has been a desired goal for computer scientists working in the music domain for the last decade. Since emotions are one of the key aspects of music perceived by listeners [99], labeling large music collections with emotions has multiple benefits in classification, navigation, recommendation, discovery of new music. However, similarly to genre detection (see Section 2.3), this task is highly complicated since there is no fixed taxonomy of emotions (see Section 4.2.1), and there is a semantic gap between human perception of music content and the way machines can represent it.

Emotion detection has been modeled as a supervised machine learning task, where the set of emotion classes is known a priori. The approaches used by researchers typically differ in the chosen machine learning technique and the set of classes (i.e., emotions). Similarly to music genre detection, achieving an overall precision higher than 60% seems to be problematic when working with large and varied datasets. The main classification approaches found in the literature include Support Vector Machines (SVM) [109,102], *k*-Nearest Neighbors (*k*NN) [110], Mixed Media Graph (MMG) [111,86], and Gaussian Mixture Models (GMM) [112,102].

Li and Ogihara [109] attempted to detect emotions in music by analyzing the audio signal. The authors addressed the problem of emotion detection as a multi-label classification task where emotions define the different classes, and music can be assigned to multiple classes. The authors used a dataset of 499 tracks belonging to 4 genres (Ambient,

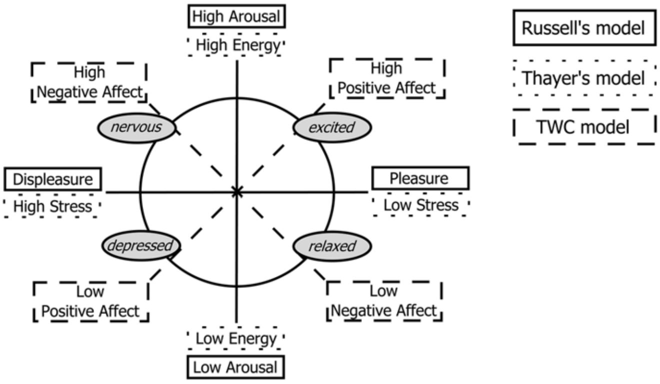

Fig. 1 – 2-dimensional emotion models.

Classical, Fusion, and Jazz). The audio features used to describe music tracks were extracted using MARSYAS audio analysis framework [113]. The features included timbre, rhythm, and pitch content. In total, each music track was represented by a 30-dimensional feature vector.

For training and testing the approach, emotions were assigned to music manually by a single person. The set of 10 emotion labels was taken from the work of Farnsworth [104] with an addition of three labels—*mysterious*, *passionate*, and *bluesy*. For experimental purposes, the 13 labels were also grouped into 6 clusters. The classification was performed using a SVM classifier. The results showed that the average accuracy does not exceed 46% when classifying into the 13 emotion classes. It reaches 50% when classifying into the 6 emotion classes. Moreover, the authors reported improvements when classifying within each genre separately.

The limitations of the proposed approach are related to the data quality. The music was labeled by one person. The tracks were represented by 30 s fragments, although the authors themselves acknowledge that a single track may contain multiple emotions. Furthermore, no justification for the selection of the audio features and the emotion groups was given.

Wieczorkowska et al. [110] used the set of 13 emotion labels and 6 clusters taken from [109]. The authors used a dataset of 875 Western music samples (each 30 s long) manually labeled by an expert. Similarly to Li and Ogihara, the detection of emotions was considered as a multi-label classification task. The authors used a set of low-level musical features that are mostly related to pitch, loudness, and the amplitude of audio signal. Each track was represented by a 29-dimensional feature vector.

Music classification into emotion categories was performed using a *k*NN approach adapted to multi-label classification. The results showed 27% accuracy when classifying music into the 13 emotion classes, and 38.6% when classifying into the 6 classes. The authors justified the low performance of their approach with the subjectivity of emotion presence in music and a large set of possible emotions. Furthermore, the selection of used low-level music features is questionable. For example, the authors did not use rhythm- and tempo-related features, but mentioned the extension of features as a future work.

Kuo et al. [111] proposed a generic model to extract emotions from music using association discovery between emotions and music features. The authors used film music for discovering the associations, assuming that the emotions of film music have been detected. By analyzing the features of film music, a model was trained to associate certain music features with emotions. Similarly to Li and Shan [86], the model used for association discovery is an adaptation of the Mixed Media Graph [97].

The used audio features are related to melody (chord progression), rhythm (repetitive beat patterns), and tempo (beat density). The authors did not rely on a particular emotion model, but used a set of emotions from the work of Reilly [114] with some additional emotions. Overall, 15 emotion labels were used. The evaluation of the approach was carried out with a set of 107 film music tracks. The results showed that the proposed approach can identify the emotions of a music track with an average precision of more than 50%. However, no comparison with existing emotion recognition approaches has been performed.

A limitation of this work is related to the reasoning behind the proposed approach. In order to predict the emotions of music tracks, a dataset of film segments labeled with emotions is needed. However, in the described study film music has been labeled with emotions manually. The authors did not address the issue of emotion extraction from movies, and relied on the reasoning that this can be done more efficiently than hand-labeling music dataset with emotions. Although the authors provided references to research on automatic detection of emotions from the visual scenes or dialogs in the movies, it is unclear whether it is feasible to fully automatize and scale the proposed approach.

Lu et al. [112] presented an approach to automatic emotion detection of classical music recordings. The authors used 4 emotion classes which correspond to the 4 quarters of the 2-dimensional Thayer's emotion model [67]—*contentment*, *depression*, *exuberance*, and *anxious/frantic*. The low-level audio signal features used to represent music tracks are related to the intensity (subband energy), timbre (spectral shape and contrast), and rhythm (rhythm strength, regularity, and tempo) of the audio signal. The authors claim that intensity features correspond directly to the arousal dimension in the Thayer's emotion model, and rhythm and timbre both correspond to the valence dimension.

The authors introduced a 2-step approach to classifying music tracks into the 4 emotion classes. During the first step, intensity features were used to classify a track into one of the two groups (*contentment/depression*, or *exuberance/anxious/frantic*). This classification task corresponds to distinguishing between the low and high arousal in Thayer's emotion model. The authors claim the arousal dimension to be more computationally tractable compared to the valence dimension. During the second step, rhythm and timbre features were used to classify the tracks in the two groups into the final 4 classes. A GMM classifier was used in both classification steps.

The evaluation of the approach was performed on a dataset of 250 classical music tracks. Multiple clips of 20 s were extracted from each track, resulting in 800 clips. The music clips were assigned to one of the 4 emotion clusters by 3 experts. The results showed some of the emotions to be easier detectable than others. For example, while the accuracy for *depression* class was 94.5%, for *contentment* it was only 76%. The average classification accuracy was 86%. Regarding the limitations of the proposed approach, the used emotion model can be mentioned. Although the authors relied on a classic Thayer's model, they chose to use only 4 emotion classes which do not cover the wide range of emotions perceived by music listeners. Therefore, although the 2-step classification approach looks promising, it should be tested with more diverse music and a more detailed taxonomy of emotions.

Dunker et al. [102] presented a framework for detecting the emotions of music and images. The authors argued that parallel emotion detection in music and images can be used in multimedia slideshow players as well as automatic tagging, navigation, and visualization of multimedia libraries.

The features used to represent music tracks include 5 low-level and 7 mid-level audio signal properties – normalized loudness, spectral features, and MFCC-based [12] timbral features – resulting in the total vector dimensionality of 219. Images were represented with 42-dimensional vectors of low-level features that covered the brightness, saturation, and color tone properties. The authors used 4 emotions corresponding to the 4 quadrants of the 2-dimensional *Valence-Arousal* model: *aggressive*, *melancholic*, *euphoric*, and *calm*.

The music dataset has been collected by querying the online music platforms (e.g., Last.fm) with emotion keywords, and afterwards manually reviewing the results by 3 persons. Similar procedure was applied to images by querying Flickr. In total, 100 tracks and images were collected for each of the 4 emotions. Two classifiers – SVM and GMM – were evaluated for the emotion detection task. The results showed that both classifiers obtain a 48.5% accuracy for emotion detection in music, and a slightly better performance for images. The authors found that for music tracks, certain emotions are more difficult to detect. For instance, *euphoric* and *calm* have proven to be more difficult to correctly detect compared to the other two emotions.

4.2.3. *Summary*

In summary, automatically detecting emotions in music is still a challenge for the MIR community. The main reasons for this are: the lack of fixed emotion taxonomy, the subjectivity of emotion perception, and the lack of direct mapping between emotions and music features.

Moreover, there is a lack of ground truth datasets for evaluation and comparison of different approaches. The training data is often hand-labeled by researchers, and each group uses its own small dataset. An attempt to produce a ground truth dataset for emotion recognition was presented by Hu et al. [115]. The authors have obtained emotion-related tags from Last.fm for 2554 tracks from the USPOP dataset.[10] Using k-means clustering, the tagged tracks were arranged into three clusters: *aggressive/angry*, *melancholic/calm*, and *upbeat/happy*. Despite being a good attempt to provide a universal dataset, this work has serious limitations: the set of music tracks is limited to pop music, and the emotion taxonomy is overly simplistic.

Another possibility to collect mood labels for music is using the tagging games [116], even though this approach has its own limitations (see Section 4.3.1).

4.3. *Music and the social web*

Social tagging, also known as collaborative tagging, was defined by Golder and Huberman [117] as "*the process by which many users add metadata in the form of keywords to shared content*". This phenomena started with the growth of online content in the late 90's [118]. The research potential of the information stored in the user generated tags was soon recognized in the Information Retrieval community. Golder and Huberman [117] analyzed the social bookmarking system Delicious[11] to understand the patterns of user activity and the tag data. The authors discovered that the types of tags used by taggers follow a power law distribution, i.e., few kinds of tags are used most frequently, while the rest form the "long tail" of the tag cloud. This is partly explained by the fact that, while annotating resources, users are often influenced by existing tags. Furthermore, the users belonging to the same tagging community have certain knowledge in common, which helps them choose similar tags.

In the field of Music Information Retrieval, Lamere [119] gave an overview of the social tagging phenomena in the context of music-related applications. The author analyzed the distribution of tags in the largest music tagging community – Last.fm –, discussed the motivations for social tagging, the different types of tagging systems, and the application of tags in MIR. The author argued that tags may become the key solution to many hard MIR problems like genre or mood detection, music discovery, and recommendation. Furthermore, Lamere identified the main problems that need to be solved before tags can be effectively used in MIR tasks: the presence of noise in tag data, lack of tags for new and unpopular items, and the vulnerability of the tagging systems to malicious users.

In this section, we provide a short overview of the research on social tagging in MIR community. We divide the discussion into two parts—first we focus on research related

[10] http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html.

[11] http://www.delicious.com.

to tagging data acquisition, e.g., tagging games or automatic tag prediction; in the second part we give some examples of tag applications for MIR tasks, such as music retrieval, discovery, and recommendation.

4.3.1. *Tag acquisition*

While the research community generally agrees that tags are a valuable source of information, a major problem recognized by researchers is the collection of high quality tag data. Considerable research efforts have been given to explore different ways of tag acquisition. The lack of tag vocabulary standards makes it difficult to define a unified tag acquisition and quality assessment strategy. Turnbull et al. [120] analyzed different ways to obtain tags for music. The authors identified five approaches: conducting surveys, harvesting social tags, deploying tagging games, mining web documents, and automatically tagging audio content. The first three methods require direct human involvement, while the last two are performed by machines. Here we briefly describe each of the five approaches.

Conducting surveys. Surveys are the most straightforward, but also the most expensive and time consuming way to obtain tags. The tag vocabulary in surveys is typically controlled and small. Having such vocabulary brings the benefit of clean and possibly structured annotations, but also results in possible loss of valuable information [119]. The participants of surveys can be experts (e.g., the Music Genome Project), or common users [121].

Large companies like Pandora and All Media Guide[12] have invested in expert studies, but do not eagerly share the outcomes of these studies with the research community. Since large scale and expert studies are expensive, researchers usually conduct tagging surveys on a small scale, and typically involve non-expert users. For instance, in some research works on the application of tagging to emotion recognition [109], or contextual music adaptation [86] small scale surveys were used to collect tag data. The number of participants in such surveys varies between 1 and a few hundreds users.

A good example of a dataset collected by means of a survey is the Computer Audition Lab 500-Song (CAL500) dataset [122,121]. 66 students were paid to listen to and annotate a set of 500 Western music songs. The tag vocabulary initially consisted of 135 labels describing the instrumentation, genre, emotion, acoustic qualities, and usage terms of the tracks. The number of labels was further increased by replacing each bipolar adjective with two individual labels (e.g., *tender* being substituted with *tender* and *not tender*). A tag has been assigned to a song if at least two users labeled that song with it. After the annotation, the least frequently used labels were removed, resulting in the final vocabulary of 174 semantic concepts. A minimum of three different tags per song were collected (1708 tags in total). The authors have conducted this survey to collect data for automatic music annotation (see below) and semantic audio retrieval (see Section 4.3.2). Moreover, the dataset has been released to the research community, and has since become a *de facto* benchmark used by other researchers when evaluating automatic annotation and retrieval algorithms (e.g., [123]).

Harvesting social tags. Social tags are accumulated in community websites that support the tagging of their content. For instance, Last.fm allows its users to freely tag tracks, artists, albums, and record labels. Lamere [119] summarized the different motivations for users to tag: tagging for future retrieval of the items, tagging for organization of the content, tagging as social signaling to other users, etc.

The major challenge faced when harvesting social tags is having the access to a sufficiently large online user community. With over 40 millions of users, Last.fm is the largest community website dedicated to music. Last.fm has millions of songs annotated with 960,000 individual tags [124]. The data is made publicly available via an API.[13]

Unlike surveys, social websites have uncontrolled vocabulary resulting in noisy data, but more diverse labels. Bertin-Mahieux et al. [123] suggested that large and noisy set of labels performs better than small and clean set when high precision is important (i.e., in recommender systems), while small and clean set is better when high recall is important (i.e., in retrieval systems).

A major problem of social tagging communities is the so-called "popularity bias" [120]: artists and songs that are most popular among the tagging users receive the majority of tags. A related problem is the "tagger bias" [119]: the harvested tags reflect the opinion of a typical "tagger" – a user who devotes his time to tagging –, which is not necessarily the opinion of the whole music lovers' community.

Deploying tagging games. Annotation games present another user-oriented approach to collecting tags. The idea of engaging users in performing large scale labeling of resources was introduced in image retrieval research. Ahn and Dabbish [125] presented "The ESP game" (ESP stands for "ExtraSensory Perception")—an online image labeling game for pairs of users. The pairs are selected randomly from all online users. The users do not know the identity of their partner nor are allowed to communicate with them. An image is displayed to both members of a pair. Each member is required to type labels that in his/her opinion the other user would use to describe the image. Once the two users type the same label, an agreement is reached on the image, and a certain number of points is assigned to the pair. The evaluation of the system has shown that while trying to think "like each other", the users produce good descriptors for the images. In subsequent years this idea has been adapted to music several times, as described in the following paragraphs.

TagATune was developed by researchers at Carnegie Mellon University. The initial design of the game [126] was based on "The ESP game": pairs of users were given an audio clip and had to label it thinking "like each other". The authors called such design of annotation games the "output agreement"—the users are given the same input (i.e., an audio clip) and have to agree on the output (i.e., annotations). After an initial evaluation the authors observed that music is not suitable for the "output agreement" games. Unlike images,

[12] http://www.allmusic.com.

[13] http://www.last.fm/api.

music has too many possible descriptors: apart from genre and instrumentation information, useful annotations may refer to the mood, temperature, the imagery created by music, etc. Consequently, in the first version of TagATune, 36% of the time the users opted to skip a track instead of providing annotations for it [126].

Based on these observations, the game was successively re-designed [127]. The new mechanism for collecting music annotations was called the “input agreement”. Differently than in “output agreement”, here the two players are not aware whether they are given the same tracks, or different ones. The players are not able to communicate, but can see each other’s annotations. Based on these they have to decide whether the track that they are listening to is the same or not. The players only get points if both guess correctly. Therefore, they are truthful to each other and provide good annotations of the tracks. Compared to the first version of the game, in the new version the players opted to pass on 0.5% rounds only. The dataset collected using TagATune was later released to public as the Magnatagatune dataset.[14]

Turnbull et al. [128] presented another music annotation game—ListenGame. The game uses the “output agreement” mechanism, where the players are asked to describe 15 s music clips by choosing the best and the worst descriptors from a list of 6 labels (randomly selected from the controlled tag vocabulary). A player is given points based on the amount of agreement between the player’s choice and the choices of other players. The usage of a controlled tag vocabulary helps the users to reach an agreement, but also limits the diversity of collected data. In order to overcome this limitation, ListenGame introduces the so-called freestyle rounds, during which the players can enter new descriptors for a clip. Unlike Law and Ahn [127], the authors of ListenGame treat the song-label associations not as binary, but as real-valued weights, proportional to the percentage of players who agree on them.

The quality of annotations collected using the ListenGame has been evaluated in a two-week pilot study. The authors used a part (250 songs) of the CAL500 dataset [121] and a subset of the CAL500 vocabulary (82 labels). The collected data was found to be sparse, since each of the 20,500 possible song-annotation pairs has been shown to taggers on average only twice. As a result, the ListenGame annotations were inferior to those of CAL500. However, the authors expected an improvement with more users playing the game. The authors of ListenGame have continued their work with a new tagging game—HerdIt [129]. It has been used by many more individuals—being deployed on Facebook, it attracted more players, and is able to incorporate the social data of the users. However, the analysis of the collected data has not been presented so far.

Kim et al. [116] developed MoodSwings—a game designed for collecting mood annotations of music. The design of the game follows the “output agreement” mechanism—the users are given the same tracks and are rewarded when their decisions match. However, differently from other tagging games, MoodSwings uses the 2-dimensional input space (Thayer’s emotion model [67]) instead of text input. Unfortunately, no evaluation of the quality of collected data has been described. That is, the obtained mood information was not compared to other existing annotations.

In conclusion, the advantage of annotation games over other human labor-based tagging methods is that the tagging cost is reduced by engaging the users in the gaming process. The task of tagging is not perceived as a job, but as a fun activity. On the other hand, the main drawback of annotation games is related to the gaming paradigm—since the primary goal of users is to maximize the score in the game, either by agreeing on as many tags as possible (i.e., the “output agreement” design), or by agreeing on as many tracks as possible (i.e., the “input agreement” design), the users tend to apply more general (“safe”) tags. Typically, the tags obtained via annotation games describe the genre and instrumental information of the tracks. Whereas more interesting tags describing, for example, the mood or the imagery of tracks are difficult to gather. Although annotation games have been successfully applied in image retrieval domain, music annotation games have not been applied on a large scale so far [120].

Mining web documents. Music tags can be automatically extracted, without explicitly asking the users to tag music, by mining the annotations for artists, albums, or tracks that are largely available in the World Wide Web. This approach is based on parsing the user-generated content in the form of music reviews, blog posts, etc. As opposed to manual annotation methods, web mining typically results in large and noisy tag vocabularies. Moreover, web annotations tend to suffer from popularity bias, as only the popular items receive sufficient attention on the web [120].

Whitman and Ellis [130] described one of the first attempts to mine the web for music annotations. The authors have mined reviews for 600 albums from two dedicated websites: All Music Guide[15] and Pitchfork Media.[16] In order to extract the important terms, TF-IDF weights were calculated for terms in the review texts. Only terms appearing in at least 3 reviews and having TF-IDF values above a certain threshold were considered for the classification task. The analysis resulted in approximately 5000 terms.

Knees et al. [131] applied web mining at the track level. For each music track in a dataset of more than 12,000 tracks, three queries were constructed and issued to Google: “<artist >music”, “<artist><album >music review”, and “<artist><title >music review -lyrics”. For each query, 100 top results were parsed and merged into a single document. Next, term weights were extracted from the text using a modification of the TF-IDF measure. This procedure resulted in approximately 78,000 distinct terms.

Having obtained the terms vector representation of each music track, the authors have used audio similarity to reduce the dimensionality of the vectors. For each track, represented by a set of MFCCs feature vectors, the 100 most acoustically similar and the 100 most dissimilar tracks were determined. These two sets have been used in a standard term selection procedure χ^2 to find n terms that best discriminate the set of

[14] http://tagatune.org/Magnatagatune.html.

[15] http://www.allmusic.com.

[16] http://www.pitchforkmedia.com.

similar tracks from the set of dissimilar tracks. After applying the χ^2 procedure to all tracks, the selected terms were merged into a global list. The authors have used n values of 50, 100, and 150, which resulted in global term lists of 4679, 6975, and 8866 respectively. Acoustic similarity of the tracks has been further used to improve the term representation of the tracks by applying smoothing of term vector weights depending on the similarity rank. This is particularly useful for tracks that do not have sufficient information on the web.

We note that the accuracy of mining web documents for music annotations is rarely evaluated as a separate task, but rather serves as the bootstrapping step for other applications, like automatic music annotation [130], or semantic music search [131]. The quality of web mining is therefore reflected in the evaluation of the final system, be it a multi-class classifier, or a music search engine.

Automatic annotation. Automatic annotation of music tracks is performed by defining a multi-class supervised machine learning task. This area of research is closely related to genre detection (see Section 2.3) and emotion recognition (see Section 4.2), since music genres and emotions are typically represented by text labels. Automatic music annotation addresses a more general problem: predicting any musically relevant labels. Thus, in automatic music annotation the set of possible classes (i.e., labels) may vary between a few (e.g., when predicting just genre or emotions), and a few hundred (when predicting more general labels). Despite the usage of supervised learning for the specific tasks of genre detection [24] and emotion recognition [109], the larger-scale task of semantic annotation of music has received little attention until recently.

For an extensive review on the state-of-the-art on automatic audio tagging refer to the work of Bertin-Mahieux et al. [132]. Here we briefly review some of the works that have contributed to this research area.

Whitman and Ellis [130] were among the first to apply machine learning for automatically annotating music tracks with meaningful labels. The authors have collected a ground truth dataset consisting of 600 albums described with approximately 5000 terms (see above for details). Music albums were represented by randomly choosing 4 songs, and extracting MFCC-based feature representation for them. For each of the terms, a binary SVM classifier has been trained. As a result, the method provided a binary decision for each term, i.e., either the term is applicable to the album, or not. The evaluation has shown the approach to perform well for some of the terms, in particular for the terms *aggressive*, *softer*, *synthetic*, *reverb*, *noise*, and *new wave*. A major limitation of this work is evaluating the approach on album level only. This limitation has been imposed by the available data: music reviews rarely describe separate songs in detail, therefore, a fine-grained training data is difficult to obtain by mining the reviews.

Turnbull et al. [122,121] proposed a generative model that produces a distribution over the vocabulary of labels for each song. The authors collected a dataset of 500 songs annotated with a vocabulary of 174 labels, known as the CAL500 dataset (see above for details). MFCCs Delta feature vectors were used to represent the audio content. For each track, 5200 39-dimensional vectors were extracted per minute of audio. These vectors were randomly sampled so that at the end each track in the dataset was represented by a set of 10,000 feature vectors. Automatic annotation was performed by modeling a distribution of each label in the tag vocabulary over the audio feature space using a Gaussian Mixture Model (GMM). Each label-level GMM was trained on the tracks associated with the label. Thus, each song could be annotated with the most likely labels.

Evaluation of the approach was performed by computing the precision and recall for each label in the vocabulary. The performance was shown to vary greatly depending on the predicted labels (as we have observed in Section 4.2.2 dedicated to automatic emotion recognition). For instance, the label "*male lead vocals*" was predicted with the precision of 96%, while some other labels, like the "*presence of a solo in a track*", had very low precision. On average, the highest precision was reached for the emotion labels (36 labels in total) that were predicted with a 42% accuracy.

Autotagger [133,123] is another system that can automatically predict social tags for music tracks based on the learned mapping between the audio features and tags mined from the web. The authors have extracted 7 million tags applied to 280,000 artists from the Last.fm folksonomy. Using the artist-level tags implies that any song by a particular artist is labeled with the artist's tags. This is a limitation (particularly for artists that create diverse music) which the authors themselves acknowledge. From a total of 122,000 unique tags, 360 most frequent ones were kept. The tags mostly describe the genre, mood, and instrumentation of the tracks.

The audio features used to describe the tracks include MFCCs [12], autocorrelation coefficients of onset trace, and spectrogram coefficients sampled over short windows (100 ms) of audio signal. In order to reduce the complexity, the features were aggregated resulting in a set of 12 features per track. Classification was done using the FilterBoost classifier [134], with each tag being learned separately.

The authors have compared Autotagger to automatic annotation approach developed by the CAL group [121]. In order to compare the two approaches, Autotagger was trained and tested on the CAL500 dataset. The obtained results were comparable, with Autotagger performing slightly better with respect to precision (i.e., predicting the correct tags for a track), and the approach of CAL group performing slightly better at recall (i.e., predicting all the relevant tags for a track).

In a recent work on autotagging, Tingle et al. [135] used a training dataset consisting of 10,870 songs (in 18 different genres) annotated with a vocabulary of 475 acoustic tags and 153 genre tags from Pandora's Music Genome Project. Each song in the dataset was annotated with appropriate genre and subgenre tags, plus 2–25 acoustic tags. The audio features representing music tracks have been obtained from the Echo Nest API.[17] The feature set – Echo Nest Timbre (ENT) – represents a track by a set of low-level feature vectors, extracted for short time frames. The ENT feature set, compared to the classical MFCC-based representation, provides more compact representation of music tracks. For instance, an average length song can be represented by

[17] http://developer.echonest.com.

several hundred ENT vectors, while the same song could have tens of thousands MFCCs vectors. The authors also tested a second feature set – Echo Nest Song (ENS) –, which represents each track by a single vector of mid-level acoustic features like beats- per-minute and average loudness.

Similarly to previous works, the authors addressed autotagging as a multi-label classification problem. A separate classifier was trained for each tag. The authors implemented three machine learning methods, previously used for autotagging by other researchers: Gaussian Mixture Model (used in [121]), Support Vector Machine (used in [130]), and Boosted Decision Stumps (used in [133]). The authors performed two experiments: comparing the two low-level feature representations (MFCCs and ENT) using the GMM classifier; and comparing ENS features (using SVM or BDS classifiers) against ENT (using the GMM classifier). The results showed ENT features to perform better than MFCCs at autotagging. The ENS features were shown to perform worse than the low-level features (ENT and MFCCs) when compared across all tags. However, for certain labels (e.g., "*triple note feel*") ENS features performed better.

All of the previously described approaches model each tag individually, without considering their relations. Duan et al. [136] exploited the relations between semantic labels to improve the performance of the automatic music annotation. For instance, the label "*hard rock*" often co-occurs with "*electric guitar*". The authors argued that incorporating this knowledge into the automatic annotation model might improve the performance. The tag vocabulary used in the study consists of 50 manually selected labels describing the genre, instrumentation, and characteristics of music signal. The authors computed the normalized mutual information of each pair of labels in the vocabulary, and selected the label pairs that have the mutual information above a certain threshold to be modeled.

The authors have used two statistical models – GMM and Conditional Random Field – in their experiments. For evaluation purpose, two versions of each model were implemented—one modeling pairs of labels, and the other modeling each label individually. Experiments were carried out on a dataset of 4951 popular songs that were manually labeled using the previously mentioned tag vocabulary. The results showed a small (up to 0.5%), but consistent improvement of precision and recall when annotating songs considering pairwise correlation of labels. The authors argued that a possible explanation for the small improvement compared to individual label modeling might occur due to the fact that in individual models, label relations are implicitly modeled (since correlated labels share many songs in the training datasets).

To summarize the discussion on automatic tagging methods, we note that to apply these methods, one needs a reliable training dataset which is difficult to generate [120]. The training data is typically obtained from surveys [121], or by mining web documents [130,123]. The former approach suffers from a severe limitation: the tag vocabulary is typically limited. The second approach has a different limitation: the vocabulary is noisy, containing many tags with minor relevance for the track. Bertin-Mahieux et al. [123] briefly discussed the advantages and disadvantages of the two types of tag vocabularies. A notable exception in terms of data quality is the dataset used by Tingle et al. [135]. Since the authors harvested Pandora's annotations done by experts, the labels are clean and consistent.

The Million Song Dataset [137] is the latest effort to produce a large-scale dataset for MIR community. The dataset was created using the Echo Nest API, and contains information on one million western commercial music tracks. Although the audio of the tracks is not available in the dataset, it contains audio features and metadata of the tracks. Unfortunately, the dataset only has tags at the artist level, which makes it difficult to use for music autotagging research. Nevertheless, the dataset is a great effort towards scaling up MIR research to realistic sizes, and it can be used for investigating a variety of research problems, such as music recommendation, cover song detection, artist recognition, etc.

4.3.2. *Tag usage for music recommendation and retrieval*

Here we review works that use social tags for MIR tasks. The applications that can benefit from annotated music content are related to music search, retrieval, and recommendation.

Music search and retrieval. Among the most popular applications for music annotations is the semantic search of music, i.e., a search service that allows users to look for music using natural language queries. Such service would be a natural way to discover music content (e.g., retrieving songs that "*have strong folk roots, feature a banjo and are uplifting*" [122]). In the research literature such retrieval is called semantic retrieval, query by text, query by semantic description, or query by description [138–140]. Although semantic retrieval has received attention in image processing research field, there has been relatively little work on this topic in MIR community. One possible reason for this is the lack of good quality large-scale datasets of annotated songs that are necessary when applying machine learning approaches to semantic retrieval.

A pioneer work on this topic is by Whitman et al. [140, 130]. The authors used a regularized least-squares classifier to learn the relation between each semantic label and the audio tracks. The used dataset consists of 255 songs by 51 artist with labels acquired automatically, by parsing web pages related to the artists (filtered to approximately 700 terms). The trained model produces a binary decision of an audio track belonging to any of the 700 output classes.

Another notable work on semantic audio retrieval was carried out by the Computer Audition Laboratory group at the University of California. The authors introduced the CAL500 dataset [122,121] (see Section 4.3.1). A GMM model was used as a multi-label classifier for both automatic annotation (see Section 4.3.1), and semantic retrieval of audio content. The results of music retrieval for single-term queries showed a mean average precision of 39% (averaged over all tags in the vocabulary of size 174). Some categories of tags proved easier to predict, and therefore performed better in the retrieval scenario. For instance, emotion queries performed with a mean average precision of 50% [121]. When evaluating the retrieval with multi-term queries, the precision dropped to 16% for 2-word queries, and to 12% for 3-word queries [122]. The authors have also used the same model with a database of 1305 sound effects annotated using a vocabulary of 348 labels, and achieved a mean average precision of 33% for retrieval of sound effects [121].

Chechik et al. [141] used a dataset of sound effects (e.g., a sound of thunder, a sound of dog barking, etc.) to develop a system that can retrieve such sound recordings from text queries. The idea of the system is to learn a mapping between textual tags and acoustic features, and then use the mapping to retrieve sounds that match a text query. The audio content was represented using standard acoustic features (MFCCs). Manually labeled training data was used for learning a matching function between the acoustic features and the labels. Three versions of learning models have been compared: GMM, SVM, and PAMIR—a model previously used for semantic retrieval of images [139]. The results showed a mean average precision of 26%–28% for all of the models with PAMIR model being considerably faster than GMM and SVM approaches, and therefore more scalable.

Knees et al. [131] used labels mined from the web (see Section 4.3.1) to implement a natural language music search engine. The authors evaluated their approach on a dataset of 12,051 music tracks. The tracks were represented by term vectors with TF-IDF weights. Queries entered by the users were expanded by the term "music" and issued to Google search engine. Top-10 results were parsed to get the final term vector representation of the query used for music retrieval.

Since ground truth datasets for semantic music retrieval are difficult to obtain, the authors used the track annotations from Last.fm dataset as a baseline in their evaluation. 227 popular Last.fm tags were used as individual search queries submitted to the search engine. For each query (i.e., tag), the returned set of tracks was compared against the tracks that have been annotated with this tag in Last.fm. Although this is not the perfect baseline for a search engine evaluation, the authors argued that Last.fm annotations are the best available source of "*phrases which are used by people to describe music and which are likely to be used when searching for music*". The results showed a precision in the range of 60%–30% at lower recall levels.

Unfortunately, research on semantic music retrieval is still in its early stages. The described music retrieval systems [121, 141,131] are only used in small scale experiments, and are not yet available for public use.

Music recommendations. Apart from being used in the MIR community, social tags are also becoming increasingly popular in the field of recommender systems (see Section 3). Tags can provide valuable knowledge about both items and users, and can be used to improve the performance of standard recommendation techniques. This is especially true for the music domain, as pure content analysis often fails to capture important aspects of human perception of music tracks.

Symeonidis et al. [142] attempted to capture the music perception of individual taggers by modeling music items, tags, and users as triplets. The triplets were stored as a 3-order tensor. Since such representation of tagging data is sparse (an average user only tags a small number of items), dimensionality reduction was applied. The authors used high order Singular Value Decomposition (SVD). Having modeled the social tagging system as a 3-dimensional dataset, the approach relies on predicting the likelihood of a user labeling a certain item with a given tag.

For the evaluation, the authors used the Last.fm dataset, with only positive tags (i.e., eliminating tags that describe negative attitude of taggers towards the items). The available dataset at the time of evaluation consisted of more than 12,000 user-tag-artist triples. The authors used a subset of this dataset by keeping those users, tags, and items that appear in at least 5 triples. Since only positive tags were used, the goal of the system is to predict any relevant tag-item pairs for a user. Intuitively, if a user is likely to assign a positive tag to a music item, this item is a relevant recommendation. The evaluation results showed the precision of around 50% at a similar recall level.

Green et al. [143] described how music tags can be used to generate transparent recommendations. The authors used a bag-of-words model to represent artists. Terms for each artist were mined from Last.fm and Wikipedia, and a weighted tag cloud representing each artist was constructed. The users can receive recommendations of similar artists that are computed using standard IR scoring techniques. These recommendations can be explained by showing the tags that contribute to computed similarity of term vectors. Moreover, such representation of artists allows the user to adjust the term weights in the cloud thus steering the recommendation process.

The authors performed an exploratory evaluation of their system by comparing it with other recommendation techniques (collaborative filtering, expert recommendation) in the task of finding similar artists based on a seed artist. While being far from a complete study, this preliminary evaluation showed that users are generally satisfied with transparent and steerable tag-based recommendations.

4.3.3. Summary

In summary, the music annotations used in research and industrial applications still vary greatly in terms of quality and quantity. The major challenges encountered when acquiring tags are: the absence of common tag vocabulary; tag polysemy and noise; choosing between a small and clean set of labels, or a large and noisy set; popularity and tagger bias; spamming [119].

The difficulty of acquiring high quality tags is the main reason why the potential of tags in music retrieval and recommendation is not yet fully exploited. A large part of the research in this area is still addressing data acquisition and cleaning methods, instead on focusing on the end-user applications. However, we believe that the future of MIR research lies within tags. Already now, tags are more than just a tool that helps users to organize and browse the online content. Tags have been shown to be useful for such MIR tasks as semantic retrieval (see above), genre and emotion detection (Sections 2.3 and 4.2), recommendation (Section 3), and context-aware adaptation (Section 4.1).

5. Conclusions

In this paper we have reviewed the state-of-the-art research on music information retrieval and recommendation. The primary goal of this review was to present a range of tools and techniques that can be used when developing context-aware music services. As the reader has noted, the research areas related to contextual music retrieval and recommendation cover a wide range of topics: signal processing, information retrieval, recommender systems, social and affective computing, cognitive psychology. Hence, it would be impossible to provide a complete review of all these areas, and this is beyond the scope of this paper. However, we provided an insight into the most important aspects of each relevant field, and we hope that this can help researchers in choosing the most appropriate techniques when addressing the challenges of context-awareness in the music domain. Efficiently using context information in music retrieval and recommendation is a challenging task, and well chosen techniques from these different fields can help to set up a solid base for starting the design of a new solution.

As said in Section 2, classical MIR techniques are based on low-level signal features which may not be enough to bring music representation closer to the music perception of humans. This problem, also known as the semantic gap [7], is also one of the obstacles in developing context-aware music services—knowing how humans perceive music is necessary when exploiting context information in a meaningful way. The subjects that may bring MIR closer to the users, and thus help developing context-aware systems include: cognitive psychology, i.e., the studies of human perception of music; affective computing, particularly emotion recognition in music; social computing, i.e., exploiting user-generated content for MIR.

Moreover, many more research studies are needed to understand what relates certain contextual conditions to music or its individual features. Although this topic has been researched in the field of music psychology [93,95,89,94,88], collaboration is needed between music psychologists and researchers in the area of music recommendations. The knowledge coming from such collaboration could help not only to improve the quality of contextual music recommendations, but also to extend the applicability of this research.

In this review we have presented topics from classical music information retrieval (MIR) and recommender system (RS) techniques along with the evolving areas of contextual music retrieval (Section 4.1), emotion recognition in music (Section 4.2), and social computing (Section 4.3).

More research results from these related topics can be found in the proceedings of the main conferences that deal with music information retrieval (ISMIR[18]), recommender systems and personalization (RecSys,[19] UMAP,[20] IUI[21]), and multimedia retrieval (ACM Multimedia[22]).

In conclusion, we hope that this survey will become a useful tool for practitioners and researchers, and will contribute to further develop the knowledge in this exciting and fast growing research area of contextual music information retrieval and recommendation.

[18] http://www.ismir.net/.
[19] http://recsys.acm.org/.
[20] http://www.um.org/.
[21] http://iuiconf.org/.
[22] http://www.acmmm11.org/.

REFERENCES

[1] F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011.

[2] G. Adomavicius, B. Mobasher, F. Ricci, A. Tuzhilin, Context-aware recommender systems, AI Magazine 32 (3) (2011) 67–80.

[3] J.Y. Kim, N.J. Belkin, Categories of music description and search terms and phrases used by non-music experts, in: 3rd International Conference on Music Information Retrieval, Paris, France, 2002, pp. 209–214.

[4] J.H. Lee, J.S. Downie, Survey of music information needs, uses, and seeking behaviours: preliminary findings, in: ISMIR Proceedings, 2004, pp. 441–446.

[5] M. Braunhofer, M. Kaminskas, F. Ricci, Recommending music for places of interest in a mobile travel guide, in: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys'11, ACM, New York, NY, USA, 2011, pp. 253–256.

[6] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lüke, R. Schwaiger, Incarmusic: context-aware music recommendations in a car, in: E-Commerce and Web Technologies—12th International Conference, EC-Web 2011, Toulouse, France, August 30–September 1, 2011. Proceedings, Springer, 2011, pp. 89–100.

[7] M.S. Lew, N. Sebe, C. Djeraba, R. Jain, Content-based multimedia information retrieval: state of the art and challenges, ACM Transactions on Multimedia Computing, Communications and Applications 2 (1) (2006) 1–19.

[8] J. Foote, An overview of audio information retrieval, ACM Multimedia Systems 7 (1998) 2–10.

[9] D.P. Ellis, Extracting information from music audio, Communications of the ACM 49 (8) (2006) 32–37.

[10] R. Typke, F. Wiering, R.C. Veltkamp, A survey of music information retrieval systems, in: ISMIR 2005, Queen Mary, University of London, 2005, pp. 153–160.

[11] P. Cano, E. Batlle, T. Kalker, J. Haitsma, A review of audio fingerprinting, The Journal of VLSI Signal Processing 41 (3) (2005) 271–284.

[12] B. Logan, Mel frequency cepstral coefficients for music modeling, in: International Symposium on Music Information Retrieval, vol. 28, 2000.

[13] E. Wold, T. Blum, D. Keislar, J. Wheaton, Content-based classification, search, and retrieval of audio, IEEE Multimedia 3 (3) (1996) 27–36.

[14] A. Wang, The shazam music recognition service, Communications of the ACM 49 (8) (2006) 44–48.

[15] W. Birmingham, R. Dannenberg, B. Pardo, Query by humming with the vocalsearch system, Communications of the ACM 49 (8) (2006) 49–52.

[16] A. Uitdenbogerd, J. Zobel, Melodic matching techniques for large music databases, in: Proceedings of the Seventh ACM International Conference on Multimedia (Part 1), ACM, 1999, p. 66.

[17] B. Pardo, M. Sanghi, Polyphonic musical sequence alignment for database search, in: Proceedings of the 6th International Conference on Music Information Retrieval, ISMIR, Citeseer, 2005, pp. 215–222.

[18] C. Meek, W. Birmingham, Automatic thematic extractor, Journal of Intelligent Information Systems 21 (1) (2003) 9–33.

[19] R. Dannenberg, W. Birmingham, B. Pardo, N. Hu, C. Meek, G. Tzanetakis, A comparative evaluation of search techniques for query-by-humming using the MUSART testbed, Technology 58 (2007) 3.

[20] A. Ghias, J. Logan, D. Chamberlin, B.C. Smith, Query by humming: musical information retrieval in an audio database, in: MULTIMEDIA'95: Proceedings of the Third ACM International Conference on Multimedia, ACM, New York, NY, USA, 1995, pp. 231–236.

[21] B. Pardo, J. Shifrin, W. Birmingham, Name that tune: a pilot study in finding a melody from a sung query, Journal of the American Society for Information Science and Technology 55 (4) (2004) 283–300.

[22] F. Pachet, D. Cazaly, A taxonomy of musical genres, in: Proc. Content-Based Multimedia Information Access, RIAO, 2000, pp. 1238–1245.

[23] N. Scaringella, G. Zoia, D. Mlynek, Automatic genre classification of music content: a survey, IEEE Signal Processing Magazine 23 (2) (2006) 133–141.

[24] G. Tzanetakis, P. Cook, Musical genre classification of audio signals, IEEE Transactions on Speech and Audio Processing 10 (5) (2002) 293–302.

[25] J.P. Bello, J. Pickens, A robust mid-level representation for harmonic content in music signals, in: Proceedings of 6th International Conference on Music Information Retrieval, ISMIR 2005, 2005, pp. 304–311.

[26] A. Rauber, E. Pampalk, D. Merkl, Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by sound similarity, in: Proc. ISMIR, 2002, pp. 71–80.

[27] P.-N. Tan, Introduction to Data Mining, Pearson Addison Wesley, San Francisco, 2006.

[28] J.G.A. Barbedo, A. Lopes, Automatic genre classification of musical signals, EURASIP Journal on Advances in Signal Processing 2007 (2007).

[29] T. Pohle, E. Pampalk, G. Widmer, Evaluation of frequently used audio features for classification of music into perceptual categories, in: Proceedings of the International Workshop on Content-Based Multimedia Indexing, 2005.

[30] R. Neumayer, A. Rauber, Multimodal Processing and Interaction: Audio, Video, Text, Springer, Berlin, Heidelberg, 2008, (Chapter) Multimodal analysis of text and audio features for music information retrieval.

[31] I. Knopke, Geospatial location of music and sound files for music information retrieval, in: Proceedings of 6th International Conference on Music Information Retrieval, 2005, pp. 29–33.

[32] I. Knopke, Sound, music and textual associations on the world wide web, in: Proceedings of 5th International Conference on Music Information Retrieval, 2004, pp. 484–488.

[33] R. Mayer, R. Neumayer, Multi-modal analysis of music: a large-scale evaluation, in: A. Rauber, N. Orio, D. Rizo (Eds.), Proceedings of the Workshop on Exploring Musical Information Spaces, ECDL 2009, Corfu, Greece, 2009, pp. 30–36.

[34] F. Pachet, J. Aucouturier, Improving timbre similarity: how high is the sky?, Journal of Negative Results in Speech and Audio Sciences 1 (1) (2004).

[35] B. Schilit, N. Adams, R. Want, Context-aware computing applications, in: Proceedings of the Workshop on Mobile Computing Systems and Applications, IEEE Computer Society, 1994, pp. 85–90.

[36] R. Picard, Affective Computing, The MIT Press, 2000.

[37] U. Shardanand, P. Maes, Social information filtering: algorithms for automating "word of mouth", in: CHI'95: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995, pp. 210–217.

[38] A. Uitdenbogerd, R. van Schnydel, A review of factors affecting music recommender success, in: Proceedings of 3rd International Conference on Music Information Retrieval, Paris, France, 2002, pp. 204–208.

[39] J. Breese, D. Heckerman, C. Kadie, et al. Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Vol. 461, 1998, pp. 43–52.

[40] F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, in: F. Ricci, L. Rokach, B. Shapira, P. Kantor (Eds.), Recommender Systems Handbook, Springer Verlag, 2011, pp. 1–35.

[41] C. Desrosiers, G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, pp. 107–144.

[42] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, ACM Press, 1994, pp. 175–186.

[43] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Analysis of recommendation algorithms for e-commerce, in: Proceedings of the 2nd ACM Conference on Electronic Commerce, ACM, 2000, p. 167.

[44] J. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, Lecture Notes in Computer Science 4321 (2007) 291.

[45] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, IEEE Computer 42 (8) (2009) 30–37.

[46] Y. Koren, R. Bell, Advances in collaborative filtering, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer Verlag, 2011, pp. 145–186.

[47] S. Funk, Netflix update: try this at home, 2006. sifter.org/simon/journal/20061211.html.

[48] R. Bell, Y. Koren, Scalable collaborative filtering with jointly derived neighborhood interpolation weights, in: ICDM, IEEE Computer Society, 2007, pp. 43–52.

[49] C. Hayes, P. Cunningham, Smart radio—building music radio on the fly, in: Expert Systems 2000, ACM Press, 2000, pp. 2–6.

[50] J. French, D. Hauver, Flycasting: on the fly broadcasting, in: Proceedings WedelMusic Conference, 2001.

[51] Q. Li, S. Myaeng, D. Guan, B. Kim, A probabilistic model for music recommendation considering audio features, Lecture Notes in Computer Science 3689 (2005) 72.

[52] I. Konstas, V. Stathopoulos, J.M. Jose, On social networks and collaborative recommendation, in: SIGIR'09: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2009, pp. 195–202.

[53] P. Lops, M. de Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, in: F. Ricci, L. Rokach, B. Shapira, P. Kantor (Eds.), Recommender Systems Handbook, Springer Verlag, 2011, pp. 73–105.

[54] M. Pazzani, D. Billsus, Content-based recommendation systems, Lecture Notes in Computer Science 4321 (2007) 325.

[55] J. Rocchio, Relevance feedback in information retrieval, in: The SMART Retrieval System: Experiments in Automatic Document Processing, vol. 313, 1971, p. 323.

[56] O. Celma, Foafing the music: bridging the semantic gap in music recommendation, in: The Semantic Web—ISWC 2006, Springer, Berlin, Heidelberg, 2006, pp. 927–934.

[57] O. Celma, Music recommendation and discovery in the long tail, Ph.D. Thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.

Please cite this article in press as: M. Kaminskas, F. Ricci, Contextual music information retrieval and recommendation: State of the art and challenges, Computer Science Review (2012), doi:10.1016/j.cosrev.2012.04.002

[58] P. Cano, M. Koppenberger, N. Wack, Content-based music audio recommendation, in: MULTIMEDIA'05: Proceedings of the 13th Annual ACM International Conference on Multimedia, ACM, New York, NY, USA, 2005, pp. 211–212.

[59] K. Hoashi, K. Matsumoto, N. Inoue, Personalization of user profiles for content-based music retrieval based on relevance feedback, in: MULTIMEDIA'03: Proceedings of the Eleventh ACM International Conference on Multimedia, ACM, New York, NY, USA, 2003, pp. 110–119.

[60] J. Foote, Content-based retrieval of music and audio, in: Multimedia Storage and Archiving Systems II, Proceedings of SPIE, 1997, pp. 138–147.

[61] D.N. Sotiropoulos, A.S. Lampropoulos, G.A. Tsihrintzis, Musiper: a system for modeling music similarity perception based on objective feature subset selection, User Modeling and User-Adapted Interaction 18 (4) (2008) 315–348.

[62] Z. Cataltepe, B. Altinel, Music recommendation based on adaptive feature and user grouping, in: Computer and Information Sciences, 2007, ISCIS 2007, 22nd International Symposium on, 2007, pp. 1–6.

[63] R. Burke, Hybrid web recommender systems, in: The Adaptive Web, Springer, Berlin, Heidelberg, 2007, pp. 377–408.

[64] J. Donaldson, A hybrid social-acoustic recommendation system for popular music, in: RecSys'07: Proceedings of the 2007 ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2007, pp. 187–190.

[65] K. Yoshii, M. Goto, K. Komatani, T. Ogata, H.G. Okuno, Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences, in: ISMIR, 2006, pp. 296–301.

[66] L. Barrington, R. Oda, G. Lanckriet, Smarter than genius? human evaluation of music recommender systems, in: Proceedings of the Tenth International Conference on Music Information Retrieval, ISMIR, 2009, Kobe, Japan, 2009, pp. 357–362.

[67] R.E. Thayer, The Biopsychology of Mood and Arousal, Oxford University Press, 1989.

[68] A.K. Dey, Providing architectural support for building context-aware applications, Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, director-Gregory D. Abowd, 2000.

[69] Q.N. Nguyen, F. Ricci, User preferences initialization and integration in critique-based mobile recommender systems, in: Proceedings of the 5th International Workshop on Artificial Intelligence in Mobile Systems, AIMS'04, Nottingham, UK, 2004, pp. 71–78.

[70] M. Weiser, Some computer science issues in ubiquitous computing, Communications of the ACM 36 (7) (1993) 75–84.

[71] B. Schilit, N. Adams, R. Gold, M. Tso, R. Want, The PARCTAB mobile computing system, in: Proceedings Fourth Workshop on Workstation Operating Systems, IEEE WWOS-IV, 1993, p. 2.

[72] J. Brotherton, G. Abowd, K. Truong, Supporting capture and access interfaces for informal and opportunistic meetings, Georgia Institute of Technology Technical Report: GIT-GVU-99 6.

[73] F. Zhou, H. Duh, M. Billinghurst, Trends in augmented reality tracking, interaction and display: a review of ten years of ISMAR, in: Proceedings of the 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality—Volume 00, IEEE Computer Society, 2008, pp. 193–202.

[74] G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, M. Pinkerton, Cyberguide: a mobile context-aware tour guide, Wireless Networks 3 (5) (1997) 421–433.

[75] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, pp. 217–253.

[76] A. Dey, G. Abowd, Towards a better understanding of context and context-awareness, in: CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness, Vol. 4, Citeseer, 2000, pp. 1–6.

[77] M. Razzaque, S. Dobson, P. Nixon, Categorisation and modelling of quality in context information, in: R. Sterrit, S. Dobson, M. Smirnov (Eds.), Proceedings of IJCAI 2005 Workshop on AI and Autonomic Communications, 2005.

[78] S. Reddy, J. Mascia, Lifetrak: music in tune with your life, in: HCM'06: Proceedings of the 1st ACM International Workshop on Human-Centered Multimedia, ACM, New York, NY, USA, 2006, pp. 25–34.

[79] H.-S. Park, J.-O. Yoo, S.-B. Cho, A context-aware music recommendation system using fuzzy Bayesian networks with utility theory, in: FSKD 2006, in: LNCS (LNAI), Springer, 2006, pp. 970–979.

[80] L. Gaye, R. Mazé, L.E. Holmquist, Sonic city: the urban environment as a musical interface, in: NIME'03: Proceedings of the 2003 Conference on New Interfaces for Musical Expression, National University of Singapore, Singapore, Singapore, 2003, pp. 109–115.

[81] J.S. Lee, J.C. Lee, Music for my mood: a music recommendation system based on context reasoning, in: P.J.M. Havinga, M.E. Lijding, N. Meratnia, M. Wegdam (Eds.), EuroSSC, in: Lecture Notes in Computer Science, vol. 4272, Springer, 2006, pp. 190–203.

[82] M. Kaminskas, F. Ricci, Location-adapted music recommendation using tags, in: J. Konstan, R. Conejo, J. Marzo, N. Oliver (Eds.), User Modeling, Adaption and Personalization, in: Lecture Notes in Computer Science, vol. 6787, Springer, Berlin, Heidelberg, 2011, pp. 183–194.

[83] A. Ankolekar, T. Sandholm, Foxtrot: a soundtrack for where you are, in: Interacting with Sound Workshop: Exploring Context-Aware, Local and Social Audio Applications, 2011, pp. 26–31.

[84] G.T. Elliott, B. Tomlinson, Personalsoundtrack: context-aware playlists that adapt to user pace, in: CHI'06: CHI '06 Extended Abstracts on Human Factors in Computing Systems, ACM, New York, NY, USA, 2006, pp. 736–741.

[85] R. Cai, C. Zhang, C. Wang, L. Zhang, W.-Y. Ma, Musicsense: contextual music recommendation using emotional allocation modeling, in: MULTIMEDIA'07: Proceedings of the 15th International Conference on Multimedia, ACM, New York, NY, USA, 2007, pp. 553–556.

[86] C.-T. Li, M.-K. Shan, Emotion-based impressionism slideshow with automatic music accompaniment, in: MULTIMEDIA'07: Proceedings of the 15th International Conference on Multimedia, ACM Press, New York, NY, USA, 2007, pp. 839–842.

[87] A. Stupar, S. Michel, Picasso—to sing, you must close your eyes and draw, in: 34th ACM SIGIR Conf. on Research and development in Information, 2011, pp. 715–724.

[88] A. North, D. Hargreaves, Situational influences on reported musical preference, Psychomusicology: Music, Mind and Brain 15 (1–2) (1996) 30–45.

[89] T. Pettijohn, G. Williams, T. Carter, Music for the seasons: seasonal music preferences in college students, Current Psychology (2010) 1–18.

[90] L. Gaye, L. Holmquist, In duet with everyday urban settings: a user study of sonic city, in: Proceedings of the 2004 Conference on New Interfaces for Musical Expression, National University of Singapore, 2004, p. 164.

[91] J. Lee, J. Lee, Context awareness by case-based reasoning in a music recommendation system, in: Ubiquitous Computing Systems, Springer, 2007, pp. 45–58.

[92] M. Zentner, D. Grandjean, K.R. Scherer, Emotions evoked by the sound of music: characterization, classification, and measurement, Emotion 8 (4) (2008) 494–521.

[93] J. Foley Jr., The occupational conditioning of preferential auditory tempo: a contribution toward an empirical theory of aesthetics, The Journal of Social Psychology 12 (1) (1940) 121–129.

[94] A. North, D. Hargreaves, The Social and Applied Psychology of Music, Cambridge Univ. Press, 2008.

[95] V. Konecni, Social interaction and musical preference, in: The Psychology of Music, 1982, pp. 497–516.

[96] M. Jones, S. Jones, G. Bradley, N. Warren, D. Bainbridge, G. Holmes, Ontrack: dynamically adapting music playback to support navigation, Personal and Ubiquitous Computing 12 (7) (2008) 513–525.

[97] J. Pan, H. Yang, C. Faloutsos, P. Duygulu, Automatic multimedia cross-modal correlation discovery, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2004, p. 658.

[98] J. Russell, A circumplex model of affect, Journal of Personality and Social Psychology 39 (6) (1980) 1161–1178.

[99] D. Huron, Perceptual and cognitive applications in music information retrieval, in: Proceedings of the 1st Annual International Symposium on Music Information Retrieval (ISMIR), 2000.

[100] Y. Kim, E. Schmidt, R. Migneco, B. Morton, P. Richardson, J. Scott, J. Speck, D. Turnbull, Music emotion recognition: a state of the art review, in: Proceedings of the 11th International Society for Music Information Retrieval Conference, 2010, pp. 255–266.

[101] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, J.G. Taylor, Emotion recognition in human–computer interaction, Signal Processing Magazine, IEEE 18 (1) (2001) 32–80.

[102] P. Dunker, S. Nowak, A. Begau, C. Lanz, Content-based mood classification for photos and music: a generic multimodal classification framework and evaluation approach, in: MIR'08: Proceeding of the 1st ACM International Conference on Multimedia Information Retrieval, ACM, New York, NY, USA, 2008, pp. 97–104.

[103] K. Hevner, Experimental studies of the elements of expression in music, The American Journal of Psychology 48 (2) (1936) 246–268.

[104] P.R. Farnsworth, The Social Psychology of Music, Iowa State University Press, 1958.

[105] E. Schubert, Update of the Hevner adjective checklist, Perceptual and Motor Skills 96 (2) (2003) 1117–1122.

[106] C. Whissell, The dictionary of affect in language, in: The Measurement of Emotion, 1989, pp. 113–131.

[107] A. Tellegen, D. Watson, L. Clark, On the dimensional and hierarchical structure of affect, Psychological Science 10 (4) (1999) 297.

[108] R. Plutchik, Emotion: A Psychoevolutionary Synthesis, Harper & Row, New York, 1980.

[109] T. Li, M. Ogihara, Detecting emotion in music, in: Proceedings of the Fourth International Conference on Music Information Retrieval, ISMIR, 2003, Baltimore, USA, 2003, pp. 239–240.

[110] A. Wieczorkowska, P. Synak, Z. Raś, Multi-label classification of emotions in music, in: Intelligent Information Processing and Web Mining, Springer, Berlin, Heidelberg, 2006, pp. 307–315.

[111] F.-F. Kuo, M.-F. Chiang, M.-K. Shan, S.-Y. Lee, Emotion-based music recommendation by association discovery from film music, in: MULTIMEDIA'05: Proceedings of the 13th Annual ACM International Conference on Multimedia, ACM, New York, NY, USA, 2005, pp. 507–510.

[112] L. Lu, D. Liu, H.-J. Zhang, Automatic mood detection and tracking of music audio signals, IEEE Transactions on Audio, Speech and Language Processing 14 (1) (2006) 5–18.

[113] G. Tzanetakis, P. Cook, Marsyas: a framework for audio analysis, Organized Sound 4 (03) (2000) 169–175.

[114] W. Reilly, Believable social and emotional agents, Ph.D. Thesis, Carnegie Mellon University, 1996.

[115] X. Hu, M. Bay, J. Downie, Creating a simplified music mood classification ground-truth set, in: Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR'07, 2007, pp. 309–310.

[116] Y. Kim, E. Schmidt, L. Emelle, Moodswings: a collaborative game for music mood label collection, in: Proceedings of the Ninth International Conference on Music Information Retrieval, ISMIR, 2008, 2008, pp. 231–236.

[117] S. Golder, B. Huberman, Usage patterns of collaborative tagging systems, Journal of Information Science 32 (2) (2006) 198.

[118] T. Hammond, T. Hannay, B. Lund, J. Scott, Social bookmarking tools (I), D-Lib Magazine 11 (4) (2005) 1082–9873.

[119] P. Lamere, Social tagging and music information retrieval, Journal of New Music Research 37 (2) (2008) 101–114.

[120] D. Turnbull, L. Barrington, G. Lanckriet, Five approaches to collecting tags for music, in: Proceedings of the 9th International Conference on Music Information Retrieval, Philadelphia, USA, 2008, pp. 225–230.

[121] D. Turnbull, L. Barrington, D. Torres, G.R.G. Lanckriet, Semantic annotation and retrieval of music and sound effects, IEEE Transactions on Audio, Speech and Language Processing 16 (2) (2008) 467–476.

[122] D. Turnbull, L. Barrington, D. Torres, G. Lanckriet, Towards musical query-by-semantic-description using the cal500 data set, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2007, pp. 439–446.

[123] T. Bertin-Mahieux, D. Eck, F. Maillet, P. Lamere, Autotagger: a model for predicting social tags from acoustic features on large music databases, Journal of New Music Research 37 (2) (2008) 115–135.

[124] F. Miller, M. Stiksel, R. Jones, Last.fm in numbers, Last.fm press material.

[125] L.v. Ahn, L. Dabbish, Labeling images with a computer game, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2004, pp. 319–326.

[126] E. Law, L. Von Ahn, R. Dannenberg, M. Crawford, Tagatune: a game for music and sound annotation, in: International Conference on Music Information Retrieval, ISMIR'07, 2007, pp. 361–364.

[127] E. Law, L. Von Ahn, Input-agreement: a new mechanism for collecting data using human computation games, in: Proceedings of the 27th International Conference on Human Factors in Computing Systems, ACM, 2009, pp. 1197–1206.

[128] D. Turnbull, R. Liu, L. Barrington, G. Lanckriet, A game-based approach for collecting semantic annotations of music, in: 8th International Conference on Music Information Retrieval, ISMIR, 2007, pp. 535–538.

[129] L. Barrington, D. O'Malley, D. Turnbull, G. Lanckriet, User-centered design of a social game to tag music, in: Proceedings of the ACM SIGKDD Workshop on Human Computation, ACM, 2009, pp. 7–10.

[130] B. Whitman, D. Ellis, Automatic record reviews, in: Proceedings of the 5th International Society for Music Information Retrieval Conference, Citeseer, 2004, pp. 470–477.

[131] P. Knees, T. Pohle, M. Schedl, G. Widmer, A music search engine built upon audio-based and web-based similarity measures, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2007, pp. 447–454.

[132] T. Bertin-Mahieux, D. Eck, M.I. Mandel, Automatic tagging of audio: the state-of-the-art, in: W. Wang (Ed.), Machine Audition: Principles, Algorithms and Systems, IGI Publishing, 2010, pp. 334–352 (Chapter 14).

[133] D. Eck, P. Lamere, T. Bertin-Mahieux, S. Green, Automatic generation of social tags for music recommendation, in: Advances in Neural Information Processing Systems, Vol. 20, MIT Press, Cambridge, MA, 2008, pp. 385–392.

[134] J. Bradley, R. Schapire, Filterboost: regression and classification on large datasets, Advances in Neural Information Processing Systems 20 (2008) 185–192.

[135] D. Tingle, Y. Kim, D. Turnbull, Exploring automatic music annotation with acoustically-objective tags, in: Proceedings of the International Conference on Multimedia Information Retrieval, ACM, 2010, pp. 55–62.

[136] Z. Duan, L. Lu, C. Zhang, Collective annotation of music from multiple semantic categories, in: Proceedings of the Ninth International Conference on Music Information Retrieval, ISMIR, 2008, 2008, pp. 237–242.

[137] T. Bertin-Mahieux, D.P. Ellis, B. Whitman, P. Lamere, The million song dataset, in: Proceedings of the 12th International Conference on Music Information Retrieval, ISMIR 2011, 2011, pp. 591–596.

[138] A.S. Chakravarthy, Toward semantic retrieval of pictures and video, in: RIAO, CID, 1994, pp. 676–687.

[139] D. Grangier, F. Monay, S. Bengio, A discriminative approach for the retrieval of images from text queries, Lecture Notes in Computer Science 4212 (2006) 162.

[140] B. Whitman, R. Rifkin, Musical query-by-description as a multiclass learning problem, in: IEEE Workshop on Multimedia Signal Processing, 2002, pp. 153–156.

[141] G. Chechik, E. Ie, M. Rehn, S. Bengio, D. Lyon, Large-scale content-based audio retrieval from text queries, in: MIR'08: Proceeding of the 1st ACM international Conference on Multimedia Information Retrieval, ACM, New York, NY, USA, 2008, pp. 105–112.

[142] P. Symeonidis, M. Ruxanda, A. Nanopoulos, Y. Manolopoulos, Ternary semantic analysis of social tags for personalized music recommendation, in: Proceedings of 9th International Conference on Music Information Retrieval, Philadelphia, USA, 2008, pp. 219–224.

[143] S. Green, P. Lamere, J. Alexander, F. Maillet, S. Kirk, J. Holt, J. Bourque, X. Mak, Generating transparent, steerable recommendations from textual descriptions of items, in: Proceedings of the Third ACM Conference on Recommender Systems, ACM, 2009, pp. 281–284.

Please cite this article in press as: M. Kaminskas, F. Ricci, Contextual music information retrieval and recommendation: State of the art and challenges, Computer Science Review (2012), doi:10.1016/j.cosrev.2012.04.002