Chapter 8

Music Data: Beyond the Signal Level

DIETMAR JANNACH, IGOR VATOLKIN
Department of Computer Science, TU Dortmund, Germany

GEOFFRAY BONNIN
LORIA, Université de Lorraine, Nancy, France

8.1 Introduction

In Chapter 5 we discussed a number of features that can be extracted from the audio signal, including rhythmic, timbre, or harmonic characteristics. These features can be used for a variety of applications of Music Information Retrieval (MIR), including automatic genre classification, instrument and harmony recognition, or music recommendation.

Beside these signal-level features, however, a number of other sources of information exist that explicitly or indirectly describe musical characteristics or metadata of a given track. In recent years, for example, more and more information can be obtained from Social Web sites, on which users can, for instance, tag musical tracks with genre or mood-related descriptions. At the same time, various music databases exist which can be accessed online and which contain metadata for millions of songs. Finally, some approaches exist to derive "high-level", interpretable musical features from the low-level signal to be able to build more intuitive and better usable MIR applications.

This chapter gives an overview of the various types of additional information sources that can be used for the development of MIR applications. Section 8.2 presents a general approach to predict meaningful semantic features from audio signal. Section 8.3 deals with features that can be obtained from digital symbolic representations of music and Section 8.4 provides a short introduction to the analysis of music scores. In Section 8.5, methods to extract music-related data from the Social Web are discussed. The properties of typical music databases are outlined in Section 8.6. Finally, Section 8.7 introduces lyrics as another possible information source to determine musical features for MIR applications.

8.2 From the Signal Level to Semantic Features

The automated classification of music and the organization of digital music collections are typically done for human listeners. It therefore seems to be helpful for users if they, for example, can understand why a set of tracks belongs to the same class. In general, to make the outcomes of an automated process better interpretable by end users, one possible goal is to derive "high-level" music data descriptors from signal-level features. Furthermore, the analysis of such interpretable features may in turn be helpful for the automated recommendation of new music, the identification of properties of certain music styles or artists, and even the automatic composition of pieces adapted to the style of a particular composer or a personal music taste, as will be discussed in Chapter 24.

8.2.1 Types of Semantic Features

The *semantic descriptors* we are interested in are typically related to music theory. Table 8.1 shows five groups of such descriptors together with examples of concrete semantic features and the related low-level signal characteristics which are often used for the estimation of the corresponding semantic descriptors. For example, features that describe inharmonic properties of semitones such as tristimulus and inharmonicity may characterize the noisiness of onsets and be helpful to recognize instruments. The chroma vector, as another example, is a basis for the estimation of key and mode (see Chapter 19 for details).

Table 8.1: Groups of Semantic Features with Examples

Group	Examples	References	Related low-level features
Instrument and vocal characteristics, playing styles, digital effects	Occurrence and share of strings in a given frame, vocal roughness	Chapter 18	Tristimulus, inharmonicity, Section 5.13
Harmony	Key and mode, chords	Chapters 3, 19	Chroma and extended variants, Section 5.3.1
Melody	Rising or falling melody, share of minor and major thirds in a melodic line, number of melodic transpositions	Chapter 3	Chroma and extended variants, Section 5.3.1
Tempo, rhythm, and dynamics	Number of beats per minute, number of bars in four-four meter, number of triplets, variance of loudness	Chapters 3, 20	Rhythmic features, Section 5.4
Emotional and contextual impact on a listener	Levels of arousal and valence; emotions fear, anger, joy, sadness; moods earnest, energetic, sentimental	Chapter 21	Root mean square energy, Equation (2.48); fluctuation patterns, Section 5.4.3

The boundaries between low-level and semantic features are often blurred. Consider, for example, the chroma as described in Equation (5.18). The idea to map all related frequencies to a semitone bin is very close to the signal level, but the progress

of a chroma component with the largest value over time may describe the melody line, which we would consider as a semantic feature based on music theory.

Generally, there exists no agreement on which features should be described as "high-level". In [5], the features are categorized by their "closeness" to a listener. Signal-level features therefore include descriptors like timbre, energy, or pitch. In contrast, rhythm, dynamics, and harmony are musical characteristics considered to be more meaningful and closer to a user. The highest-level features according to [5] are referred to as "human knowledge" and relate to the personal music perception (emotions, opinions, personal identity, etc.). These features are particularly hard to assess. In yet another categorization scheme, [39] describes seven "aspects" of musical expression: temporal, melodic, orchestrational, tonality and texture, dynamic, acoustical, electromusical and mechanical. Rötter et al. [35] finally list 61 high-level binary descriptors suitable for the prediction of personal music categories.

8.2.2 Deriving Semantic Features

Many semantic features can be directly estimated from the digital score as will be discussed in the next section. However, the score might not always be available. For audio recordings, supervised classification methods – including those introduced in Chapter 12 – can be applied to derive semantic features. To train the classification models, ground truth labels are required for the classification instances (typically frames of time signal) which are represented by features. For example, the labels may indicate the occurrence of a particular instrument or a mood in the frame. This information can be provided by music experts or collected from web databases like The Echo Nest or AllMusicGuide; see Section 8.6.

Supervised classification can be applied in an incremental manner, where already calculated characteristics are used to predict the next ones. This approach is similar to classification chains proposed in [31], where the result of a classification model becomes itself a feature for the prediction of additional classes. An individual model in such an approach would predict, e.g., a mood or the occurrence of a particular instrument.

A general procedure called *Sliding Feature Selection* (SFS) [42] is sketched in Figure 8.1. Here, in each step, classification models are built (preferably with an ensemble of classifiers), and for each model only the most relevant features are kept after multi-objective feature selection, which minimizes the number of features and the classification error simultaneously.[1] The number of features on a level i is given by N_i. Note that on each level the new features do not replace the previous ones but extend the pool of available descriptors. For a better interpretability of the final models, it can then be reasonable to remove the low-level signal features in the last training step.

Not all possible sequences of extraction steps are, however, meaningful. For example, we may expect that temporal and rhythmic properties do not necessarily

[1] Multi-objective optimization is introduced in Section 10.4, multi-objective feature selection in Section 15.7, and classification methods in Chapter 12.

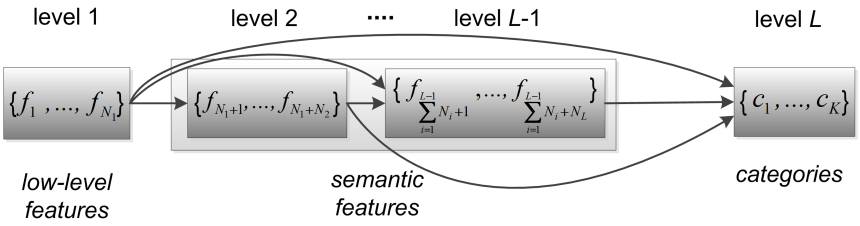

Figure 8.1: Sliding Feature Selection.

improve or simplify the recognition of instruments. Example 8.1 shows several possible sequences of SFS. The final step in each sequence obviously comprises the prediction of the aspect that we are actually interested in.

Example 8.1 (Levels of Sliding Feature Selection).

- *low-level features* $\mapsto$ *instruments* $\mapsto$ *moods* $\mapsto$ *genres*
- *low-level features* $\mapsto$ *instrument groups (keys, strings, wind)* $\mapsto$ *individual instruments* $\mapsto$ *styles* $\mapsto$ *genres* $\mapsto$ *personal preferences*
- *low-level features* $\mapsto$ *harmonic properties* $\mapsto$ *moods* $\mapsto$ *styles*
- *low-level features* $\mapsto$ *harmonic properties* $\mapsto$ *rhythmic patterns* $\mapsto$ *moods* $\mapsto$ *styles* $\mapsto$ *genres*

The individual levels of the SFS chain (cf. Figure 8.1) can be combined with *feature construction* techniques; see Section 14.5. In the described generic approach, new features can be constructed through the application of mathematical operators (like sum, product, or logarithm) on their input(s). As an example, consider a chroma vector which should help us identify harmonic properties. In the first step, new characteristics can be constructed by summing up the strengths of the chroma amplitudes for each pair of chroma semitones. The "joint strength of C and G" – as a sum of the amplitudes of C and G – would, for example, measure the strength of the consonant fifths C-G and fourths G-C. The overall number of these new features based on 12 semitone strengths is equal to $\frac{1}{2} \cdot 12 \cdot 11 = 66$. In the next step, the "strength of sad mood" could be predicted after applying the SFS chain using a supervised classification model which is trained on the subset of these 66 descriptors that leads to the best accuracy.

8.2.3 Discussion

The extraction of robust semantic characteristics from audio signals of several music sources is generally challenging, both in cases where various signal transforms (see Chapters 16-21) are applied or when supervised classification is used as introduced above. Nevertheless, the selection of the most relevant semantic features can be helpful to support a further analysis by music scientists or help music listeners to understand the relevant properties of their favorite music.

The usage of higher-level descriptors derived with the help of SFS may not necessarily improve the classification quality when compared to approaches that only use low-level features, as both methods start with the same signal data. Another challenge is the proper selection of training data in each classification step. For example, too many training tracks from the same album typically lead to a so-called *album effect*, where the characteristics of albums are learned instead of genres. In the context of the genre recognition problem, however, SFS was proven to be sufficiently robust in [44] and the experiments also showed that models that were trained on a subset of the features performed significantly better than models that relied on all available features.

8.3 Symbolic Features

In Chapter 7, the MIDI format was presented as one way of digitally and symbolically encoding music in a structured "how to play" form. In the MIR literature, several approaches exist that try to derive or reconstruct musical features from the MIDI encoding and use them in different applications. Note that *symbolic features* can be extracted from various digital formats. In this section, we, however, restrict our examples to MIDI files because of their popularity.

In [22], for example, the goal was to automatically determine the musical style for a given MIDI file, since style information is commonly used to classify and retrieve music. In their multi-step approach, the authors first propose a method to extract or approximate the main melody, which is not always trivial because there can be multiple channels, i.e., multiple notes sound at the same time. In the second step, chords are assigned to the melody based on music theoretical considerations. Finally, the resulting melody and chord patterns are matched with a set of classification rules that were learned using a larger set of training data.

Music classification based on melody lines was also the goal of the work proposed in [8] where hidden Markov models were trained on a set of folk songs from different countries. In contrast to [22], only monophonic melodies were considered.[2] In [12], the authors experimented with various machine learning approaches for musical style recognition based on MIDI files. Finally, in [24] the authors analyzed MIDI-encoded musical pieces with respect to several parameters including pitch, pitch distance, duration or melodic intervals or melodic bigrams and trained artificial neural networks for tasks such as author attribution or style identification.

Unlike the previous works, Cataltepe et al. in [4] first transform the MIDI files into audio and then combine the extracted audio features with the MIDI features for genre classification. To use the MIDI features for classification, they are first automatically extracted and then transformed into a string representation, based on which the similarity of two musical pieces can be determined [9].

Generally, a large number of musical features can be extracted from MIDI files. In [25], for example, 109 different features were determined and used for a genre

[2] The data files used in the experiments used two special symbolic formats. Using MIDI-encoded files would be possible in principle.

classification task based on a combination of neural networks and a k-nearest neighbors method (cf. Section 12.4.2). Their feature set covered aspects like instrumentation, musical texture, rhythm, melody, chords and others.

In [26], the jSymbolic software library was presented that can extract 160 different "high-level" features[3] from MIDI files. The more recent *music21* toolkit [11] is even capable of determining more than 200 features, supports various input formats, and is thereby able to process features that cannot be captured in the MIDI format, for example, enharmonic tones.

Given such a large set of features, the problem can arise, however, that some classification techniques do not work very well anymore ("curse of dimensionality") as the number of required labeled cases increases strongly. Possible ways of mitigating this problem suggested by the authors of [26] include the manual or automated selection of features (see Chapter 15) based on the application domain or the construction of intermediate representations such as histograms from which further features can be derived (see e.g., [41]).

Example 8.2 (Extraction of Symbolic Features for Classical, Pop, and Rock Pieces). *A set of 12 features from jSymbolic is provided in Table 8.2. The features were extracted for two classical pieces (Cla1: Bach, Toccata and Fuga in D minor, BWV 565; Cla2: Beethoven, Sonata in C sharp minor 'Moonlight', Op. 27 No. 2), two pop pieces (Pop1: Abba, Thank You for the Music; Pop2: Madonna, Hung Up), and two rock pieces (Roc1: Nightwish, Stargazers; Roc2: Scorpions, Wind of Change).*

We can observe that some of the features may help to identify a genre. Both classical pieces are characterized by a higher level of chromatic motion, rather rising melodic intervals, a higher fraction of tritones, and a higher variability of note duration. Pop tracks have a high fraction of octaves and rock tracks a positive fraction of electric guitar. Both pop and rock pieces have a larger amount of arpeggiation. Other features like rhythmic properties or the importance of the bass register seem to be less relevant. Note that in this example the number of tracks is very low. A reliable analysis of genre properties should be done with a significantly larger number of MIDIs.

By some authors, using features extracted from MIDI files is considered easy when compared to situations when only the audio signal is available [45]. For instance, some interpretable music characteristics like instruments or harmonic and melodic properties can be directly extracted from the score. This may be very hard for polyphonic audio recordings.

On the other side, symbolic formats also have their limitations. For new or less popular music pieces, the score may be not available, and it is harder to extract style properties of a concrete performer. Detecting higher-level musical structures or musical aesthetics as discussed in [23] and [24] can be challenging. The MIDI format is also not suited to express nuances of musical scores as mentioned in [11] such as the detection of enharmonic tones or the difference between an eighth note and a

[3] Those are features that considered to be "musical abstractions that are meaningful to musically trained individuals."

Table 8.2: Examples of Features from jSymbolic, Alphabetically Sorted

Feature	Cla1	Cla2	Pop1	Pop2	Roc1	Roc2
Amount of arpeggiation (fraction of related horizontal intervals)	0.467	0.545	0.484	0.655	0.728	0.577
Chromatic motion (fraction of melodic intervals corresponding to a semitone)	0.109	0.106	0.078	0.024	0.062	0.020
Combined strength of the two strongest rhythmic pulses	0.028	0.329	0.484	0.262	0.199	0.189
Direction of motion (fraction of melodic intervals that are rising rather than falling)	0.470	0.529	0.353	0.431	0.332	0.509
Electric guitar (fraction)	0	0	0	0	0.239	0.191
Importance of bass register (fraction of notes between MIDI pitches 0 and 54)	0.175	0.329	0.095	0.489	0.664	0.236
Melodic octaves (fraction)	0.088	0.056	0.074	0.150	0.048	0.068
Melodic tritones (fraction)	0.031	0.023	0.024	0.000	0.006	0.003
Pitch variety (number of pitches used at least once)	57	60	62	48	52	49
Repeated notes (fraction)	0.039	0.192	0.079	0.364	0.576	0.129
Rhythmic variability (standard deviation of bin values)	0.019	0.026	0.032	0.021	0.022	0.015
Variability of note duration (standard deviation, in s)	0.855	0.752	0.694	0.470	0.332	0.734

staccato quarter. Therefore, the authors of [11], for example, propose to combine MIDI features with other features, including lyrics, popularity information, or chord annotations that can be obtained from different sources.

8.4 Music Scores

In Section 8.3 we have seen that a form of a "digital score" like MIDI allows us to do various types of automated analysis like melody extraction, which in turn help us build more elaborate solutions, e.g., for music classification. There might, however, be situations where only the (printed) music score is available instead of a digital symbolic representation of the music. In order to exploit the information from the score in a music data analysis scenario, it is therefore necessary to visually analyze the music sheet, recognize the various symbols, and store them in a machine-processable form like MIDI.

The automated recognition of printed sheet music has been investigated by researchers for decades. Some early works in *optical music recognition* (OMR)[4] date back to the late 1960s as discussed, for example, in the survey of Carter et al. from 1988 [3]. At first glance, the problem appears to be a comparably simple document analysis problem, because the set of symbols are defined, there are staff lines, and there are some quite strict rules that can be used to validate and correct the hypotheses that are developed during the recognition process [34]. In practice, however, OMR is considered to be challenging because, for example, the individual symbols

[4]Other terms are *optical score reading* or *music image analysis*.

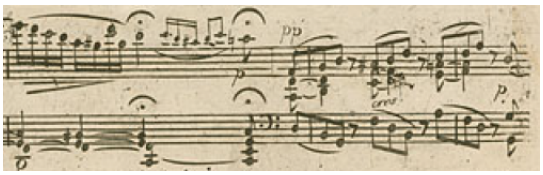

Figure 8.2: Fragment of a printed and scanned score.

can be highly interconnected (see Figure 8.2) and that they can vary in shape and size even within the same score [33].

An OMR process usually consists of several phases [32]. First, image processing is done, which involves techniques such as image enhancement, binarization, noise removal or blurring. The second step is symbol recognition, which typically includes tasks like staff line detection and removal, segmentation of primitive symbols and symbol recognition, where the last step is often done with the help of machine learning classifiers which are trained on labeled examples. In the following steps, the identified primitive symbols are combined to build the more complex musical symbols. At that stage, graphical and syntactical rules can be applied to validate the plausibility of the recognition process and correct possible errors. In the final phase, the musical meaning is analyzed and the symbolic output is produced, e.g. in terms of a MIDI file.

Over the years, a variety of techniques have been proposed to address the challenges in the individual phases, but a number of limitations remain in particular with respect to hand-written scores. At the same time, from a research and methodological perspective, better means are required to be able to compare and benchmark different OMR systems [32].

From a practical perspective, today a number of commercial OMR tools exist including both commercial ones like SmartScore[5] and open-source solutions like Audiveris.[6] According to [32], these tools produce good results for printed sheets, but have limitations when it comes to hand-written scores.

8.5 Social Web

During the last decade, *Social Web* platforms have become popular and nowadays link millions of users. Several of these social platforms support a number of social interactions about music which can be used for music data analysis tasks. These interactions, for example, include the collaborative annotation of music through *tags*, sharing of hand-crafted *playlists*, or the recording, publication and discussion of the users' music *listening activities*.

[5] `http://www.musitek.com`. Accessed 03 January 2016

[6] `https://audiveris.kenai.com`. Accessed 03 January 2016

8.5.1 Social Tags

One common feature provided by music websites like Last.fm is to let users assign tags to musical resources. Usually, such tags are freely chosen by the users and can be, for instance, the genre of an artist, the mood of a track, the year of release of an album, etc.

As these music websites are visited by millions of users, the number of tags available on these sites can be much higher than the amount of music annotations that could be done by music experts. Moreover, as these tags are assigned in a collaborative way, the subjectivity of each individual annotation can at least partially result in "inter-subjective" annotations.

However, as tags are freely chosen by non-expert users, they usually contain a lot of noise. For instance, tracks can be tagged with advertisements for other online services, or are misused by the users as a bookmark tool if the website allows to search music by tags. This noise can be ignored if a sufficient number of different users have tagged a given track. Unfortunately, tags tend to be concentrated on the most popular tracks [6]. This makes it difficult to use tags as an additional source of information for less popular or new music [7]. For more information about how social tags can be collected and used see [18] and [40].

8.5.2 Shared Playlists

Another particular source of knowledge about music are *playlists* that are created and shared by the users of music platforms. Websites and platforms allowing users to create and share playlists include Last.fm, 8tracks,[7] Art of the Mix,[8] and Spotify.[9]

One interesting piece of information contained in such playlists are relationships between tracks which were made by the playlist creators but cannot be captured solely from metadata or the audio signal. For instance, if two tracks are found one after the other in several playlists, then it can be deduced that both these tracks share something important, even if their content and metadata are completely different. These relationships can, of course, also correspond to the content and metadata, which can also be interesting, particularly when the content or metadata are not known. For instance, playlists often group tracks of the same genre, and this information can be used to infer the genres of the tracks.

The extraction of relationships between tracks from playlists is often based on the co-occurrences of the tracks (or artists). This, however, means that to be reasonably confident about the relationship between two tracks (or artists), they must appear together in a sufficient number of playlists. Therefore, this strategy allows us to capture only limited information for the less popular tracks, in particular when compared to what can be obtained for the same tracks based on their content or metadata.

In the following, we present an approach from [43] to derive genre information that works reasonably well even when the analyzed tracks occur seldom in a large

[7] `http://www.8tracks.com`. Accessed 03 January 2016
[8] `http://www.artofthemix.org`. Accessed 03 January 2016
[9] `https://www.spotify.com`. Accessed 03 January 2016

set of playlists, see Example 8.3. To measure the "degree of co-occurrence" of two artists, the concepts of *support* and *confidence* from the field of association rule mining can be used.

Vatolkin et al. in [43] define the normalized support $Supp(a_i, a_j)$ of two artists a_i and a_j in a collection of playlists P as the number of playlists in which both a_i and a_j appeared divided by the number of playlists $|P|$. The confidence value $Conf(a_i, a_j)$ relates the support to the frequency of an artist a_i, which helps to reduce the overemphasis on popular artists that comes with the support metric.

Let us now assume that our problem setting is a binary classification task with the goal to predict the genre of an unknown artist (or, analogously, the genre of a track where we know the artist). We assume that each artist is related to one predominant genre. Our training data can therefore be seen to contain T_p annotated "positive" examples of artists for each genre ($ap_1, ..., ap_{T_p}$) and T_n artists who do not belong to a given genre ("negative" artists $an_1, ..., an_{T_n}$). To learn the classification model we now look at our playlists and determine for each "positive" artist ap_i those artists that appeared most often together with ap_i. Similarly, we look for co-occurrences for the negative examples for a given genre.

When we are now given a track of some artist a_x to classify, we can determine with which other artists a_x co-occurred in the playlists. Obviously, the higher the co-occurrence of a_x with artists that co-occurred also with some ap_i, the higher the probability that a_x has the same predominant genre as ap_i. Otherwise, if a_x often co-occurs with artists that do not belong to the genre in question, we see this as an indication that a_x does not have this predominant genre either. Technically, the co-occurrence statistics are collected in the training phase and used as features to learn a supervised classification model (see Chapter 12). In [43], experiments with different classification techniques were conducted and the result showed that the approach based on playlist statistics outperformed an approach based on audio features for 10 out of 14 tested genres. The results also showed that using *confidence* is favorable in estimating the strength of a co-occurrence pattern in most cases.

Example 8.3 (Extraction of Artist Co-Occurrences in Playlists). *Table 8.3 shows those five artists (provided in the table header) that most frequently co-occur with four "positive" artists for the genres Classical, Jazz, Heavy Metal, and Progressive Rock based on Last.fm playlist data.*

Even if the top co-occurring artists are very popular, this method can be helpful to classify less popular artists. For example, after the comparison of support values for Soulfly*, the most probable assignment would be the genre Heavy Metal given the statistics of the data set used in [43]: $Supp($Soulfly, Beethoven$) = 5.841$E-5, $Supp($Soulfly, Miles Davis$) = 2.767$E-5, $Supp($Soulfly, Metallica$) = 516.539$E-5, $Supp($Soulfly, Pink Floyd$) = 66.810$E-5.*

One underlying assumption of the approach is that playlists are generally homogeneous in terms of their genre. Also, this method does not take into account that artists can be related to different genres over their career. Finally, a practical challenge when using public playlists is that artists are often spelled differently or even wrongly, consider, e.g., "Ludwig van Beethoven", "Beethoven", "Beethoven,

Table 8.3: Top 5 Co-Occurrences for Artists with the Predominant Genres Classical, Jazz, Heavy Metal, and Progressive Rock

Chopin, Frederic	Baker, Chet	AC/DC	The Alan Parsons Project
Beethoven, Ludwig van	Davis, Miles	Metallica	Pink Floyd
Bach, Johann Sebastian	Simone, Nina	Iron Maiden	Genesis
Mozart, Wolfgang Amadeus	Holiday, Billie	Guns'n'Roses	Queen
Radiohead	Coltrane, John	Led Zeppelin	Dire Straits
Tchaikovsky, Pyotr	Fitzgerald, Ella	The Beatles	Supertramp

Ludwig van", "L.v.Beethoven", etc. A string distance measure can help to identify identical artists, when the distance is below some threshold. In [43], for example, the Smith–Waterman algorithm [36] is applied to compare artist names.

Generally, shared playlists can be used for music-related tasks other than genre classification. They can, for instance, provide a basis for automated playlist generation and next-track music recommendation; see for example [1], [16] and Chapter 23.

8.5.3 Listening Activity

Some of the today's Web music platforms record the details of the tracks that are played by their users. For instance, the users of Last.fm can let the system record the artist name, title, and timestamp of each track they played, which is referred to as "scrobbling".

The resulting data is called the listening logs and can be exploited to derive various types of information. One possible type of useful information, e.g. for music recommendation, is the popularity of the tracks (or artists), which can be calculated simply by counting the overall number of occurrences of the tracks (or artists) in the logs. Similarly, the currency of a track can be computed by also exploiting the timestamps.

Another example of information contained in the logs is the listening duration for each track. This information can be used to determine whether a track was played entirely or "skipped" [2]. Again, in a playlist generation scenario, these skips can represent a negative signal regarding the compatibility of two tracks and an automated playlisting application can try to avoid such patterns. Note that the hand-crafted playlists discussed in the previous section do not contain such negative information.

However, as the only available information is some track identifier (e.g. the artist and track names) and a timestamp, it is impossible to be sure that a track was fully played because the user enjoyed it or if it was played because the user was busy and could not click on the "skip" button. It is also impossible to know that a track was skipped because the user thought it did not fit to the previous track or if it was skipped because the user simply wanted a change of atmosphere. For more information about how the listening activity of users can be collected and analyzed; see [2] and [29].

8.6 Music Databases

Over the last years, a number of free and commercial music databases have become available on the Web. These databases, which can usually be accessed programmatically via standardized Web interfaces, contain a variety of information that can be used in music-related applications like music recommendation, playlist generation or the structuring of music collections. The existing music databases can be categorized along different dimensions.

- Creation and maintenance: Some music databases like *MusicBrainz* are created and curated by music enthusiasts, others like *Gracenote* are maintained by commercial service providers and major music labels.[10]
- Genre scope: Some databases are devoted to very specific musical genres like Heavy Metal[11] or Latin music [15]. Others cover a broad spectrum and provide information about millions of musical tracks.
- Content: Most databases focus on artist and basic track metadata like duration, release date, chart positions, as well as community-provided information like tags. A few databases like *The Echo Nest*[12] contain information about musical features like the (average) tempo, energy or loudness of the individual tracks. Some databases like *AllMusic*[13] also provide mood annotations and community ratings; see also Table 21.6 in Chapter 21.

Today's music databases can be huge. *Gracenote*, for example, as of 2014 claims to have information about more than 180 million different tracks in their database. On *The Echo Nest*, details for over 35 million tracks can be accessed and for many of them, detailed musical features are available. Finally, even the community-curated *MusicBrainz* website and database hosts information of about 13 million tracks.

From the MIR perspective, the database by *The Echo Nest* is probably the most interesting one as it contains – beside the track and artist metadata mentioned above – detailed information about features to which one would otherwise only have access after a computationally intensive extraction phase. The features extracted from the audio signal include, for example, the duration, begin and end of fade-in and fade-out parts, the mode, loudness, segment information, MFCCs, or the tempo. Most of these feature values are accompanied by a confidence value. From the audio-based signals, a number of additional features are derived using an internal logic including "danceability", energy, or "acousticness".

Finally, many of the mentioned music databases and Web platforms provide a number of additional functionalities that can be used when developing music applications. Typical features include the automated generation of playlists from seed songs, the calculation of tracks, artists, or genres that are similar to a currently played one, automatic recognition of tracks based on sound samples, or a service for correcting artist misspellings.

[10] `http://www.musicbrainz.org`, `http://www.gracenote.com`. Accessed 03 January 2016
[11] `http://www.metal-archives.com`. Accessed 03 January 2016
[12] `http://the.echonest.com`. Accessed 03 January 2016, acquired by Spotify in 2014
[13] `http://allmusic.com`. Accessed 03 January 2016

Apart from these public music databases and services, the database used by *Pandora*,[14] the probably most popular Internet radio station in the United States at the moment, is worth mentioning. The Internet radio is based on the data created in the *Music Genome Project*. In contrast to databases which derive features from audio signals, each musical track in the *Pandora* database is annotated by hand by musical experts in up to 400 different dimensions ("genes").[15] The available genes depend on the musical style and can be very specific like "level of distortion on the electric guitar" or "gender of the lead vocalist".[16] The annotation of one track is said to last 20 to 30 minutes; correspondingly, the size of the database – approximately 400,000 tracks – is limited when compared to other platforms.

8.7 Lyrics

Many music tracks, particularly in the area of popular music, are "songs", i.e., they are compositions for voice and performed by one or more singers. Correspondingly, these tracks have accompanying *lyrics*, which in turn can be an interesting resource to be analyzed and used for music-related applications. For example, instead of trying to derive the general mood of a track based only on the key or tempo, one intuitive approach could be to additionally look at the lyrics and analyze the key terms appearing in the text with respect to their sentiment.

In the literature, a number of approaches exist that try to exploit lyric information for different MIR-related tasks. In [14], for example, the authors combine acoustic and lyric features for the problem of "hit song" prediction. Interestingly, at least in their initial approach, the lyrics-based prediction model that used Latent Semantic Analysis (LSA) [13] for topic detection was even slightly better than the acoustics-based one; the general feasibility of hit song prediction is, however, not undisputed [30].

Also the work of [21] is based on applying an LSA technique on a set of lyrics. In their work, however, the goal was to estimate artist similarity based on the lyrics. While the authors could show that their approach is better than random, the results were worse than those achieved with a similarity method that was based on acoustics, at least on the chosen dataset. Since both methods made a number of wrong classifications, a combination of both techniques is advocated by the authors.

Instead of finding similar artists, the problem of the Audio Music Similarity and Retrieval task in the annual Music Information Retrieval eXchange (MIREX) is to retrieve a set of suitable tracks, i.e., a short playlist, for a given seed song. In [20], the authors performed a user study in which the participants had to subjectively evaluate the quality of playlists generated by different algorithms. Several participants of the study stated that they themselves build playlists based on the lyrics of the tracks or liked certain playlists because of the similarity of the content of their lyrics. This indicates that lyrics can be another input that can be used for automated playlist generation. As lyrics alone are, however, not sufficient and other factors like track

[14] `http://www.pandora.com`. Accessed 03 January 2016

[15] `http://www.pandora.com/about/mgp`. Accessed 03 January 2016

[16] `http://en.wikipedia.org/wiki/Music\_Genome\_Project`. Accessed 03 January 2016

popularity have to be taken into account, lyrics-based features have to be combined with other inputs, e.g. in a faceted scoring approach as proposed in [16].

Experiencing music is strongly connected to emotions – as discussed in depth in Chapter 21 – and automated mood detection (classification) is a central task in Music Information Retrieval. Some works try to determine the mood of musical tracks with the help of their lyrics [10, 47]. Instead of an LSA technique, the authors of these works use Term-Frequency / Inverse Document Frequency (TF-IDF) representations of the lyrics as an input to their mood classification tasks. TF-IDF representations are commonly used for document retrieval tasks in the Information Retrieval Literature. The idea is to determine importance weights for the (subset of relevant) terms appearing in a document, resulting in TF-IDF vectors. The weights are determined by multiplying two factors. The Term-Frequency component TF assigns higher scores to terms that appear more often in a document, assuming that these words are more important. The IDF component assigns higher values to terms that appear infrequently in the whole document corpus, assuming that rarely used words are more discriminative than others.[17]

Table 8.4 shows an example of TF-IDF vectors for three Christmas-related pop songs. The term "christmas" obtains very high weights for the given song collection because the term is occurring several times in each track (TF weight) and at the same time is only rarely used in all other songs (IDF weight). Term vectors like these can then be used for different MIR-related purposes. For example, they can serve as feature vectors in a mood classification problem.

Alternatively, the angle between two vectors (cosine similarity) can be used to retrieve similar tracks for a given seed track. Other similarity measures are discussed in Section 11.2. The examples in Table 8.4 show that in the retrieval scenario a few overlapping terms like "snow" can be sufficient to retrieve tracks that have at least some similarity with a seed track. Tracks that have no word in common will be considered to be completely unrelated.[18]

Table 8.4: Example for TF-IDF Vectors

Terms/Track	christmas	feed	...	bell	everyday	snow
Do they know it's Christmas	0.863	0.379	...	0.057	0.000	0.054
I wish it could be Christmas everyday	0.736	0.000	...	0.197	0.400	0.140
Let it snow	0.000	0.000	...	0.000	0.000	0.862

TF-IDF Calculation Details [17]

The calculation of the TF-IDF vectors for a collection of text documents d typically begins with a pre-processing step. In our case, each document contains

[17] Mathematically, different ways to compute the weights are possible. For an example, see [10].

[18] Compared to Latent Semantic Analysis techniques mentioned above, TF-IDF-based approaches cannot uncover hidden (latent) relationships between terms.

the lyrics of one track. In this phase, irrelevant so-called "stop-words" like articles are removed. Furthermore, *stemming* can be applied, a process which replaces the terms in the document with their word stem.

We then compute a normalized term-frequency value $TF(i,j)$, which represents how often the term i appears in document j. Normalization should be applied to avoid that longer text documents lead to higher absolute term-frequency values. Different normalization schemes are possible. For instance, we can compute the normalized frequency value of a term by dividing it by the highest frequency of any other term appearing in the same document. Let $maxFrequencyOtherTerms(i,j)$ be the maximum frequency of terms other than i appearing in document j. If $freq(i,j)$ represents the unnormalized frequency count, then

$$TF(i,j) = \frac{freq(i,j)}{maxFrequencyOtherTerms(i,j)}. \tag{8.1}$$

The IDF component of the TF-IDF encoding reduces the weight of a term proportional to its appearance in documents across the entire collection. Let N be the number of documents in d and $n(i)$ be the number of documents in which term i appears. We can calculate the Inverse Document Frequency as

$$IDF(i) = log\frac{N}{n(i)} \tag{8.2}$$

and the final TF-IDF score as *TF-IDF(i,j)*$= TF(i,j) \cdot IDF(i)$.

The resulting term vectors can be very long and sparse as every word appearing in the documents corresponds to a dimension of the vector. Therefore, additional pruning techniques can be applied, e.g., by not considering words that appear too seldom or too often in the collection.

An approach that combines lyric and acoustic information is presented in [37]. In this work, the application scenario is to identify and retrieve musical tracks based on the user's singing voice. In contrast to previous approaches that only rely on melody identification (as done in "query by humming" approaches), the authors first try to recognize the lyrics and identify the track based on the lyrics. In a second step, melody information is extracted to verify the lyrics-based retrieval result and to thereby further increase the retrieval accuracy. A similar "query-by-singing" approach was later proposed in [28], which was, however, not combined with an acoustic retrieval method.

Finally, a few works exist that aim at the automatic transcription of lyrics from the audio signal, e.g., [28]. The problem is often considered to be challenging because of the polyphonic background music and the differences between spoken and sung voices as mentioned in [27]. One particular problem in that context is the detection of *phonemes* (a "unit of speech" in a language) as basic building blocks for the lyric transcription problem. A comparison of using different supervised classification techniques and different features sets for this task can be found in [38].

Overall, lyrics have been successfully used as an add-on information source in various MIR applications, including mood and emotion detection, see [46] or [19], song classification and identification or hit song prediction. Given the recent developments in the area of sentiment analysis and the increasing availability of lyric databases as well as "ground truth" information about moods, e.g., on the *AllMusic* platform and other music databases, further advances can be expected in the area.

8.8 Concluding Remarks

In this chapter, we reviewed a variety of different types of information and data sources that can be applied in music data analysis tasks. In particular the increasing availability of public music databases and the collective knowledge available on Social Web platforms will, in our view, open a variety of new opportunities in the future to end up, e.g., with better music recommendation and music classification techniques.

Bibliography

[1] G. Bonnin and D. Jannach. Automated generation of music playlists: Survey and experiments. *ACM Computing Surveys*, 47:1–35, 2014.

[2] K. Bosteels, E. Pampalk, and E. E. Kerre. Evaluating and analysing dynamic playlist generation heuristics using radio logs and fuzzy set theory. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 351–356. International Society for Music Information Retrieval, 2009.

[3] N. Carter, R. Bacon, and T. Messenger. The acquisition, representation and reconstruction of printed music by computer: A review. *Computers and the Humanities*, 22(2):117–136, 1988.

[4] Z. Cataltepe, Y. Yaslan, and A. Sonmez. Music genre classification using midi and audio features. *EURASIP Journal of Applied Signal Processing*, 2007(1):150–150, 2007.

[5] Ò. Celma and X. Serra. FOAFing the music: Bridging the semantic gap in music recommendation. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):250–256, 2008.

[6] Ò. Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, 2010.

[7] Ò. Celma and P. Lamere. Music recommendation tutorial. International Society for Music Information Retrieval Conference (ISMIR), September 2007.

[8] W. Chai and B. Vercoe. Folk music classification using hidden Markov models. In *Proc. of the International Conference on Artificial Intelligence (ICAI)*, Las Vegas, 2001.

[9] R. Cilibrasi, P. Vitányi, and R. De Wolf. Algorithmic clustering of music based on string compression. *Computer Music Journal*, 28(4):49–67, 2004.

[10] H. Corona and M. P. O'Mahony. An exploration of mood classification in the million songs dataset. In *Proc. of the 12th Sound and Music Computing Conference (SMC)*. Music Technology Research Group, Department of Computer Science, Maynooth University, 2015.

[11] M. S. Cuthbert, C. Ariza, J. Cabal-Ugaz, B. Hadley, and N. Parikh. Hidden beyond MIDI's reach: Feature extraction and machine learning with rich symbolic formats in music21. In *Proc. of the NIPS 2011 Workshop on Music and Machine Learning*, 2011.

[12] R. B. Dannenberg, B. Thom, and D. Watson. A machine learning approach to musical style recognition. In *Proc. of the International Computer Music Conference (ICMC)*, pp. 344–347. Michigan Publishing, 1997.

[13] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[14] R. Dhanaraj and B. Logan. Automatic prediction of hit songs. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, pp. 488–491, 2005.

[15] C. L. dos Santos and J. Silla, Carlos N. The Latin music mood database. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):1–11, 2015.

[16] D. Jannach, L. Lerche, and I. Kamehkhosh. Beyond "hitting the hits": Generating coherent music playlist continuations with the right tracks. In *Proc. of the 9th ACM Conference on Recommender Systems (RecSys)*, pp. 187–194, New York, 2015. ACM Press.

[17] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, 2011.

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

[19] C. Laurier, J. Grivolla, and P. Herrera. Multimodal music mood classification using audio and lyrics. In *Proc. of the 7th International Conference on Machine Learning and Applications (ICMLA)*, pp. 688–693. IEEE Computer Society, 2008.

[20] J. H. Lee. How similar is too similar?: Exploring users' perceptions of similarity in playlist evaluation. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 109–114. University of Miami, 2011.

[21] B. Logan, A. Kositsky, and P. Moreno. Semantic analysis of song lyrics. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME)*, volume 2, pp. 827–830. IEEE, 2004.

[22] S. Man-Kwan and K. Fang-Fei. Music style mining and classification by melody. *IEICE Transactions on Information and Systems*, 86(3):655–659, 2003.

[23] B. Manaris, T. Purewal, and C. McCormick. Progress towards recognizing and classifying beautiful music with computers: MIDI-encoded music and the Zipf-Mandelbrot law. In *Proc. of IEEE SoutheastCon 2002*, pp. 52–57. IEEE, 2002.

[24] B. Manaris, J. Romero, P. Machado, D. Krehbiel, T. Hirzel, W. Pharr, and R. B. Davis. Zipf's law, music classification, and aesthetics. *Computer Music Journal*, 29(1):55–69, 2005.

[25] C. Mckay and I. Fujinaga. Automatic genre classification using large high-level musical feature sets. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, pp. 525–530, 2004.

[26] C. Mckay and I. Fujinaga. jSymbolic: A feature extractor for MIDI files. In *Proc. of the International Computer Music Conference (ICMC)*, pp. 302–305. Michigan Publishing, 2006.

[27] M. McVicar, D. Ellis, and M. Goto. Leveraging repetition for improved automatic lyric transcription in popular music. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pp. 3117–3121. IEEE, 2014.

[28] A. Mesaros and T. Virtanen. Automatic recognition of lyrics in singing. *EURASIP J. Audio Speech Music Process: Special Issue on Atypical Speech*, 2010:4:1–4:7, January 2010. `http://dx.doi.org/10.1155/2010/546047`.

[29] J. L. Moore, S. Chen, D. Turnbull, and T. Joachims. Taste over time: The temporal dynamics of user preferences. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 401–406, 2013.

[30] F. Pachet and P. Roy. Hit song science is not yet a science. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, pp. 355–360, 2008.

[31] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In *Proc. of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Part II*, pp. 254–269, 2009.

[32] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Maral, C. Guedes, and J. S. Cardoso. Optical music recognition: State-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, 2012.

[33] F. Rossant. A global method for music symbol recognition in typeset music sheets. *Pattern Recognition Letters*, 23(10):1129–1141, 2002.

[34] F. Rossant and I. Bloch. A fuzzy model for optical recognition of musical scores. *Fuzzy Sets and Systems*, 141(2):165–201, 2004.

[35] G. Rötter, I. Vatolkin, and C. Weihs. Computational prediction of high-level descriptors of music personal categories. In B. Lausen, D. van den Poel, and A. Ultsch, eds., *Algorithms from and for Nature and Life*, pp. 529–537. Springer, 2013.

[36] T. Smith and M. Waterman. Identification of common molecular subsequences.

Journal of Molecular Biology, 147:195–197, 1981.

[37] M. Suzuki, T. Hosoya, A. Ito, and S. Makino. Music information retrieval from a singing voice using lyrics and melody information. *EURASIP Journal of Applied Signal Processing*, 2007(1), 2007.

[38] G. Szepannek, M. Gruhne, B. Bischl, S. Krey, T. Harczos, F. Klefenz, C. Dittmar, and C. Weihs. *Classification as a Tool for Research*, chapter Perceptually Based Phoneme Recognition in Popular Music, pp. 751–758. Springer, 2010.

[39] P. Tagg. Analyzing popular music: Theory, method and practice. *Popular Music*, 2:37–65, 1982.

[40] D. Turnbull, L. Barrington, and G. R. Lanckriet. Five approaches to collecting tags for music. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, volume 8, pp. 225–230, 2008.

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

[42] I. Vatolkin. *Improving Supervised Music Classification by Means of Multi-Objective Evolutionary Feature Selection*. PhD thesis, Department of Computer Science, TU Dortmund, 2013.

[43] I. Vatolkin, G. Bonnin, and D. Jannach. Comparing audio features and playlist statistics for music classification. In A. F. X. Wilhelm and H. A. Kestler, eds., *Analysis of Large and Complex Data*, pp. 437–447, 2016.

[44] I. Vatolkin, G. Rudolph, and C. Weihs. Evaluation of album effect for feature selection in music genre recognition. In *Proc. of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 169–175, 2015.

[45] C. Xu, M. Maddage, X. Shao, F. Cao, and Q. Tian. Musical genre classification using support vector machines. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pp. 429–432. IEEE, 2003.

[46] D. Yang and W.-S. Lee. Music emotion identification from lyrics. In *Proc. of the 11th IEEE International Symposium on Multimedia (ISM)*, pp. 624–629. IEEE Computer Society, 2009.

[47] M. Zaanen and P. Kanters. Automatic mood classification using TF*IDF based on lyrics. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 75–80. International Society for Music Information Retrieval, 2010.