J Intell Inf Syst
DOI 10.1007/s10844-016-0430-7

Genre classification of symbolic pieces of music

Marcelo G. Armentano[1] · **Walter A. De Noni**[2] · **Hernán F. Cardoso**[2]

Received: 28 April 2016 / Revised: 22 August 2016 / Accepted: 9 September 2016

Abstract Automatic classification of music is a complex and interesting research problem due to the difficulties that arise when determining the musical features that should be considered for classification and the characteristics that define each particular genre. In this article, we propose an approach for automatic genre classification of symbolic music pieces. We evaluated our approach with a dataset consisting of 225 pieces using a taxonomy of three genres and nine subgenres. Results demonstrate that by only extracting a small set of features from the MIDI files, we are able to obtain better results than competing approaches that use one hundred features for classification.

Keywords Music genre classification · Symbolic music classification · MIDI

1 Introduction

Music Information Retrieval (MIR) is a research area that has received special attention in the last two decades (Wieczorkowska and Ras 2003; Schedl et al. 2014a). This research area is concerned with the extraction and inference of meaningful features from music, the indexation of music using these features, and the development of applications for searching and retrieving music (Downie 2003).

Due to the increasing number of music available nowadays, it becomes necessary to find alternative ways of managing and accessing all the existing information. This is especially

✉ Marcelo G. Armentano
marcelo.armentano@isistan.unicen.edu.ar

[1] ISISTAN Research Institute (CONICET / UNICEN), Campus Universitario. Paraje Arroyo Seco, Tandil, Argentina

[2] Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires, Tandil, Argentina

Published online: 22 September 2016

true for music providers and online music libraries that manage large amount of files for which the success is highly dependent on the ease with which users are able to find the kind of music they like. In this context, users can search the available music using different criteria (author, recording year, genre, etc), the genre being one of the most subjective high level features. Most of these sites currently rely on a manual classification by genre, a methodology that is slow and error prone. For this reason, automatic classification of music into genres is a necessary and complex task.

Music genre classification is also useful in other domains, such as recommender systems, in which genres can be an additional feature to be considered for building the users' profiles by learning about the kind of music they prefer, or to automatically creating different playlists. For example, streaming applications such as Spotify,[1] Last.fm[2] and Pandora[3] recommend related artists. Websites such as Musicovery[4] and Musicroamer[5] allow users to discover new music by using a visualization approach connecting artists in a graph like structure. Musicovery, in addition, distributes songs in four dimensions related to mood (Dark vs. Positive and Calm vs. Energetic) and uses different colors to indicate the genre.

The purpose of this work is to study the classification of songs into different genres from the point of view of content. Nowadays, music is generally stored in digital audio format (wav, aiff, mp3, etc.) or symbolic format (MIDI, GUIDO, MusicXML or Hundrum). The audio format is a digital representation of analog waves. The symbolic format stores musical events and parameters instead of waves. The symbolic representation is also called "high level" representation while the audio format "low level" representation. In general, symbolic representation stores data such as the duration of notes and tone, among other events.

Both kinds of representations of digital music have their advantages and disadvantages. The first difference is related to timbral features. For example, MIDI files store references to standard MIDI synthesizers' patches rather than the sound itself, with the result of losing much timbral information. The ability to extract information such as the quality of the singing voice, the lyrics, the phrasing and expression is limited when working from MIDI files (McKay 2004). This is a potentially serious problem and there is some evidence that certain features of the timbre may be more significant than other features based on the rhythm or tone (Tzanetakis and Cook 2002). However, other study on the relationship between gender and timbre indicates that there may be a poor correlation between these features (Aucouturier and Pachet 2003), indicating that the loss of timbre information may not be as important as it seems at a first sight. An additional problem with MIDI files is that this format was designed mainly based on the model of Western music, which limits its use beyond this paradigm.

On the other hand, it is very difficult to extract from an audio file format some high-level features, such as the duration of notes and individual pitches (de Jesus Guerrero-Turrubiates et al. 2014; Li and Yang 2015). This is not a problem with symbolic music, where it is easier to access to high-level information of the piece of music.

MIDI files also have the additional advantage that, given the nature of the data and the speed with which it can be processed, the complete piece of music can be used to extract

[1] https://www.spotify.com/

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

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

[4] http://musicovery.com/

[5] http://musicroamer.com/

features. In contrast, most research using the audio format only use a fragment of the file for features extraction.

Last but not least, another benefit of using symbolic data format is that it is possible to classify music for which we only have the score but not an audio version available. Optical music recognition is a very active research area and there exist many approaches to obtain a MIDI file by scanning a music sheet (Benetos et al. 2013; Chen and Sheu 2014; Wen et al. 2015). Our research seeks in part to show the importance of high-level features for music classification. The high level features have the advantage of having musical meaning, so they can be used for theoretical studies as well as for other applications.

The rest of this paper is organized as follows. In Section 2, we define the concept of music genre, and discuss how they can be classified. Next, in Section 3 the relationship between genres and music features is discussed, reviewing some features used previously to model music genres. This discussion continues in Section 4, were we present some related work on music genre classification. Section 5 presents the approach proposed in this article for music genre classification. We describe the transformation process from a MIDI file to the final structures required by our classifiers, and the classification process itself. In Section 6 we present the results obtained and compare them to a competing approach. Finally, in Section 7 we present our conclusions and future work.

2 Music genres

The concepts "music genre" and "music style" are often confusing. Fabbri (1999) defines a music genre as "*a kind of music, as it is acknowledged by a community for any reason or purpose or criteria, i.e., a set of musical events whose course is governed by rules (of any kind) accepted by a community*". In contrast, a music style is defined by Fabri as "*A recurring arrangement of features in musical events which is typical of an individual (composer, performer), a group of musicians, a genre, a place, a period of time*". Music genres can then be considered as something wider and more subjective than a music style, from the point of view of content.

Then, a music style is more related to the conventions of music features or properties (such as melody, rhythm, harmony, instrumentation, arrangement, etc) that might be associated with music from a particular region, artist or genre. The concept of genre, on the other hand, has a more taxonomical meaning, and group music into structurally related categories. As an example, a composer can create music of different genres, but his/her style might be unique. In other words, we can say that the term genre can be understood as a reference to music that, by means of a social consensus, can be grouped together.

Since music is not science, there is always a subjective component when classifying music into genres. Furthermore, the boundaries among genres are not sharp but fuzzy. We humans are able to identify genres by comparing the features we perceive in a piece of music with the cultural background and musical knowledge we have related to different music genres. For example, at a very high level, we can distinguish between instrumental and vocal music. Then, most people are able to identify different broad categories, such as rock, pop, jazz/blues, western classical, latin, etc. Depending on the musical background and knowledge, some people are also able to identify different subgenres. For example, some people are able to distinguish between jazz and blues, while other people might classify songs belonging to these genres into the same category. Furthermore, there are songs that lie at the boundary between two or more genres, and its classification is difficult both for humans and an automatic system.

There is another mechanism by which we can identify the genre of a song and is by its similarity with another song for which we know the music genre. For example, we might not know the genre of the song "Blue 'N Boogie" by Dizzy Gillespie, but we can find it similar to "The cooker" by George Benson and therefore say that it belongs to the Bebop genre.

In order to train a system to perform an automatic classification of music into genres using supervised learning, we need to know beforehand the set of genres into which the training set can be partitioned. There is no consensus about a static and ideal set of music genres. Music genres are not defined only by unchanging features, but are the result of a dynamic cultural process and therefore both its composition and definition change with time (McKay 2004). What we currently known as "rock and roll", for example, was once classified as "new"; recordings by Bob Marley that we now know as "reggae" were once known as "folk". A labeling scheme for music genres should be kept updated since it must conform the labels used in the present.

On the other hand, genres are not isolated clusters of songs, and there often exists a hierarchical relationship among them. Besides the problems described above for classifying a song into top level genres (that arise also for subgenres classification), we have the problem of deciding to which extent a genre ramifies into subgenres. Some genres, such as Country music, have only a limited set of specialized subgenres (Bluegrass, Contemporary, Traditional Country, etc). On the other hand, other genres, such as Jazz, tend to have more specific subgenres.

Some previous works on music genres classification only consider a limited number of simple unrelated music genres. In this article, we reported the classification results both at leaf level (subgenres) and at root level (genres). One of the main advantages of working with genres and subgenres is that in situations in which a piece of music cannot be classified into a subcategory with enough confidence, the upper category to which it belongs may be predicted more accurately.

3 Genres and music features

The first step in the task of music genres classification is the extraction of features from the music file that describes the latent genre. Features that are commonly extracted in the literature can be grouped into timbral texture, rhythmic, pitch content and combinations of them (Tzanetakis and Cook 2002). The number of features extracted also plays an important role in the classification performance. Too few features may fail to efficiently codify all the information that is necessary to describe each particular genre, while too many features may increase the system complexity due to excessive computation.

The identification of what features define a musical genre is a difficult task, even for the human ear. Many important attributes of music that determines the musical genre are related not only to the content of the music itself, but to cultural aspects that cannot be extracted from musical files.

It is accepted in the literature that the features selected to represent a piece of music do not need to have a theoretical unified sense. Then, genre-specific features could be used to determine the genre according to the presence or absence of each feature (for example, swing rhythm, specific percussion set, etc). The main disadvantage of basing the features on each particular genre is that this solution is very hard to maintain if the genres' taxonomy changes. Furthermore, as discussed in Section 2, there is no consensus on what features defines a particular genre. For the reasons exposed before, we choose to use features that are common to a wide set of genres.

We put special attention to simple features that a trained human being could be able to use to determine a genre. Since we humans are actually the most expert classifiers, we believe that we should use features that are easily recognized by humans. This hypothesis does not imply that other features are not important for determining the music genre, but we try to demonstrate that we can get good classification performance by using only a short set of features that are natural for humans.

In general, each author proposes a different set of features to describe pieces of music. Tagg (1982) proposed the following *checklist of parameters of musical expression* that can be adapted to extract features both from audio and symbolic music recordings:

- *Temporal aspects*: duration of the object of analysis and its relationship with any other simultaneous forms of communication; duration of sections within the object of analysis; pulse, tempo, metre, periodicity; rhythmic texture and motifs.
- *Melodic aspects*: register; pitch range; rhythmic motifs; tonal vocabulary; contour; timbre.
- Orchestational aspects: type and number of voices, instruments, parts; technical aspects of performance; timbre; phrasing; accentuation.
- Tonality and texture aspects: tonal centre and type of tonality (if any); harmonic idiom; harmonic rhythm; type of harmonic change; chordal alteration; relationships between voices, parts, instruments; compositional texture and method.
- Dynamic aspects: levels of sound strength; accentuation; audibility of parts.
- Acoustical aspects: characteristics of the place where it is performed; reververation; distance between sound source and listener; simultaneous extraneous sound.
- Electromusical and mechanical aspects: panning, filtering, compressing, phasing, distortion, delay, mixing, etc.; muting, pizzicato, tongue flutter, etc.

As stated before, the problem with extracting large sets of features is the impact on the performance of the classifier, both on training and in classification time. In this article we demonstrate that by only using three high-level features, we obtain results close to those obtained using more and more complex features.

4 Related work on music genre classification

In this section, we review some related work in musical genres classification of symbolic pieces of music.

One of the first approaches of symbolic music classification was presented by Dannenberg et al. (1997). This approach, tested with trumpet recordings in MIDI format, distinguishes in real time among eight different execution styles using different classification schemes: Bayesian, lineal and neural networks. The Bayesian classifier obtained the best performance (around 90 %).

Chai and Vercoe (2001) used Hidden Markov models to classify monophonic melodies belonging to three different kinds of occidental folk music (Austrian, German and Irish). Authors obtained a success rate of 63 % using only melodic features. An interesting finding in this research is that the number of hidden states only has a minor effect on the success rate and the simplest model outperformed the most complex models.

Ponce de León and Iñesta (2002) presented an approach that extracts and segments monophonic tracks of classical and jazz music to extract melodic, harmonic and rhythmic features. These features are used to create categories using self organized maps, obtaining a success rate close to 77 %. Similarly, Shan et al. (2002) extracted a set of features

based only on melody and chords information. Authors distinguish among four categories (Enya, Beatles, Chinese folk and Japanese folk) obtaining a success rate between 64 % and 84 %.

Wojcik and Kostek (2010) presented an approach to create *hypermetric* rhythm on the basis of both monophonic and polyphonic melodies in the MIDI format. The rhythm structure obtained is then used to create automatic drum accompaniment to a given melody. The main features used by this approach include the pitches of sounds, onsets and durations of sounds, sequence of frequencies, sequence of interfals, sequence of approximate intervals and sequence of directions of interfals.

Abeßer et al. (2012) only focused on the bass track to explore stylistic similarities between music genres. Authors presented an extensive set of transcription-based high-level features related to rhythm and tonality that allows the characterization of bass lines on a symbolic level. They verified that several known stylistic relationships between music genres by classifying typical bass lines and obtained an accuracy of 64.8 % for genre classification based only on this track.

Approaches reviewed so far, only consider one-track music files (monophonic). Regarding the classification of multi-track files, McKay (2004) developed a tool for automatic classification of pieces of music in symbolic format by combining classifiers based on neural networks and *k*-nearest neighbors (KNN) and selection and weighting features using genetic algorithms. This research proposes a wide library of features based on instrumentation, texture, rhythm, tone, melody and chords. In Section 6, we use this tool to compare the results obtained with the approach proposed in this paper.

Kotsifakos et al. (2013) proposed a novel similarity measure for sequences, called SMBGT, with the KNN classifier. This approach first extracts the channels of MIDI files and then transforms each channel into a sequence of 2D points, providing information for pitch and duration of notes. The SMBGT for all pairs of two given songs is computed to determine their distance (similarity) between them. Finally, the genre of a given song is determined by the genre of the majority of the *k* nearest neighbors. This research reported an accuracy of 40 % for the classification of songs into four genres.

Valverde-Rebaza et al. (2014) explored features from symbolic MIDI representation, such as histograms of notes, statistical moments and structural features derived from

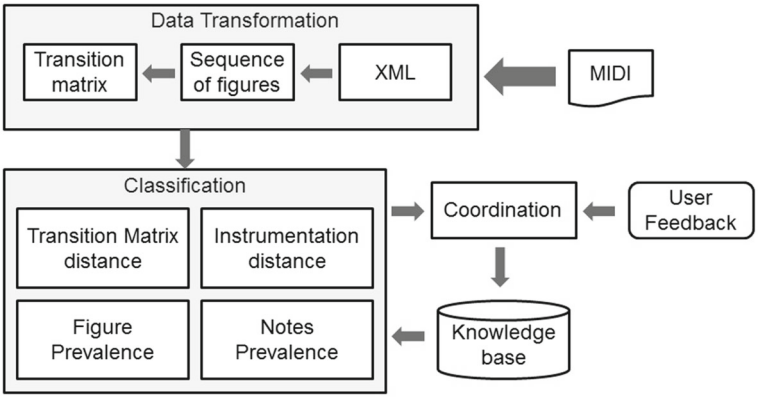

Fig. 1 Overview of the proposed approach

concepts of music theory, for genres classification using different relational classifiers. The best performance was achieved by regular-kNN graphs.

Schedl et al. (2014b) proposed a social media approach for estimating music similarity based on contextual co-occurrences of music artists harvested from microblogs. Authors of this research concluded that collaborative chatter on music can be effectively used to develop music artist similarity measures. These findings could be also used for music genre classification, based on artists' similarity, provided that the artist is bounded to one genre or very similar genres.

In the next section, we propose our proposed approach for symbolic music classification into genres. Next, in Section 6, we evaluate our approach and compare the results obtained with the results given by McKay (2004) approach.

Table 1 MIDI to XML transformation

MIDI	XML
0 Meta TrkName	<Event>
"Happy Birthday"	<Absolute>0</Absolute>
	<TrackName>HappyBirthday</TrackName>
	</Event>
0 Meta Text "trad."	<Event>
	<Absolute>0</Absolute>
	<TextEvent>trad.</TextEvent>
	</Event>
0 TimeSig 3/4 24 8	<Event>
	<Absolute>0</Absolute>
	<TimeSignature Numerator="3" LogDenominator="2"
	MIDIClocksPerMetronomeClick="24"
	ThirtySecondsPer24Clocks="8" />
	</Event>
0 KeySig 255 major	<Event>
	<Absolute>0</Absolute>
	<KeySignature Fifth="255" Mode="0"/>
	</Event>
0 Tempo 600000 512	<Event>
KeySig 255 major	<Absolute>0</Absolute>
	<SetTempo Value="600000"/>
	</Event>
513 Meta TrkEnd	<Event>
	<Absolute>512</Absolute>
	<KeySignature Fifth="255" Mode="0"/>
	</Event>
	<Event>
	<Absolute>513</Absolute>
	<EndOfTrack/>
	</Event>

5 Proposed approach

The proposed approach for symbolic music classification is shown in Fig. 1.

The knowledge base is implemented using MongoDB (from "humongous", meaning "huge"). MongoDB is a NoSQL database that instead of storing data using tables, it uses JSON (Javascript Object Notation) document with a dynamic scheme (called BSON or binary JSON) that makes the integration of data with applications easy and fast. MongoDB calls each record or set of information "document". Documents can be grouped in collections that are equivalent to tables of relational databases, with the difference that collections can store documents with different format, not restricted to a fixed scheme.

Given as input a musical piece in MIDI format, the classification process is divided in four steps, which are detailed in the following subsections:

1. Data Transformation: different features are extracted from the musical files
2. Classification: each classifier assigns a candidate genre based on the computation of the nearest neighbor by comparing the knowledge base with the transformation done to the musical file to be classified
3. Coordination: by considering the confidence on each classifier, a voting scheme is used to assign a genre to the given musical file
4. User feedback: the knowledge base is updated using the user feedback In the remaining of this Section we give details about each component of Fig. 1

5.1 Data transformation

In this sub section, we describe the process in which the musical file in symbolic format is transformed into a set of features needed for training the classifiers.

The process starts by transforming a musical file in MIDI format into an intermediate XML format. A MIDI file is composed of segments. The file starts with a header segment, followed by one or more track segments. The header segment contains information regarding the file content while the track segments contain the MIDI messages or events. There are as many track segments as instruments in the piece of music. Table 1 shows an example transformation for the MIDI file "Happy birthday".

5.1.1 XML to sequence of figures

The next transformation consists in obtaining a sequence of figures for each musical instrument in the piece of music in the XML representation. Table 2 shows the XML tags considered for this transformation.

The algorithm searches for the events of interest at each track in the piece of music. The events of interest are those represented by the tags <Program Change> (PC), <Note on> (Non) and <Note off> (Noff). For these events, the algorithm checks whether the channel property is equals to 10 to determine if it corresponds to the percussion channel. Figure 2 shows the algorithms for building the sequence of figures for each instrument.

If the tag is <Program change> and the channel is different than 10, the algorithm initializes the instrument.

If the tag is <Note on>, the algorithm checks for the Velocity property. If Velocity is 0, then the event is equivalent to <Note off> and the algorithm computes the percussion figure or the instrument figure. The Velocity is different than 0, the instrument or percussion is processed, depending on the value of the property channel.

Table 2 XML tags considered in the construction of the sequences of figures

Tag	Value	Description	Property	Value	Description
<Format>	0	The musical piece contains only one track			
	1	The musical piece contains multiple tracks			
<TicksPerBeat>	N	Duration of a quarter note			
<Track>	N	Delimits the tracks. It is composed of a collection of musical and control events			
<Absolute>	N	Indicates the moment when an event occurs (applies to all tags next)			
<ProgramChange>		Indicates the activation of an instrument	Channel	0-15	Channel number. Channel 10 is the percussion channel
			Number	0-127	Instrument number
<Note On>		Activation of a musical note	Channel	0-15	Channel number
			Note	N	Musical Note
			Velocity	N	Volume (if equals to 0 has the same effect than Note Off)
<Note Off>		Deactivation of a musical note	Channel	0-15	Channel number
			Note	N	Musical Note
			Velocity	0	Volume

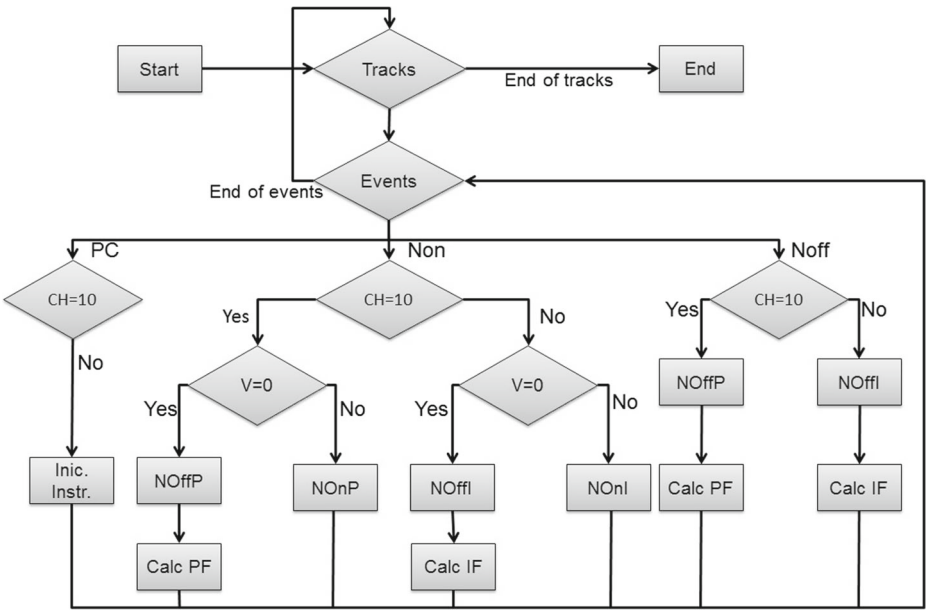

Fig. 2 Algorithm for building the sequence of figures from the xml representation

Finally, if the tag is <Note off>, the algorithm only verifies the value of the Channel property to determine if it should calculate percussion or instrument figure.

Instrument initialization (Inic. Instr.) When observing a ProgramChange event, the algorithm obtains the Number attribute, corresponding to the instrument, and verifies whether it is already registered in the corresponding data structure. The instruments data structure contains the following information, for each instrument:

- Stack of events: registers the NoteOn events for a given musical note. When the corresponding NoteOff event is observed (or, equivanlently NoteOn with Velocity=0) the event is unstacked.
- Instrument time: used to compute the duration of rests. When a NoteOn tag is observed, there will be an associated Absolute tag within the Event tag indicating the moment in which the event occurred. The difference between instrument time and the Absolute value of this event is computed, giving the duration of the previous rest.
- Instrument rest: used as an indicator of rest (flag). When the stack of events is empty, it means that the current instrument is not longer playing and the flag is activated.

Note Off Percussion (NOffP) and Note Off Instrument (NOffI) The corresponding instrument entry is recovered from the instruments data structure. The last NoteOn event for the same note is recovered and unstacked from the stack of events and the duration is computed as the difference of the Absolute values. With this duration the figure is determined. If the stack of events is empty after unstacking the event, the rest flag for the instrument is activated. Finally, the Absolute value corresponding to the NoteOff event is copied to the variable Instrument time.

Note On Percussion (NOnP) and Note On Instrument (NOnI) The corresponding instrument entry is recovered from the instruments data structure. The NoteOn event with the corresponding Absolute value is stacked in the stack of events. If the rest flag is on, the rest duration is computed, the flag is set to off and the instrument time variable is updated with the Absolute value

Computation of the percussion and instrument figure (CalcPF and CalcIF) The difference between the ending time (Absolute value of the NoteOff event being processed) and the starting time (obtained from the stack of events) is computed. Considering that T is the value of the tag <TicksPerBeat> that corresponds to the duration of a quarter figure, the figure is computed according to the values presented in Table 3.

Computation of rests The difference between the ending time (Absolute value of the NoteOn event being processed) and the starting time (equals to the instrument time) is computed. Considering that T is the value of the tag <TicksPerBeat> that corresponds to the duration of a quarter figure, the rest figure is computed according to the same values presented in Table 3.

Unification of different instances of the same instrument The result of the algorithm presented in Fig. 2 is a set of sequences of figures, one for each musical instrument present in the piece of music being modelled. However, in a given piece of music there may exist more than one instances of the same instrument (for example, two acoustic guitars). In the case of format 1 MIDI files, these instruments will be playing in different tracks, while in the case of format 0 MIDI files, in different channels. In both cases, this situation will generate two or more independent sequences for the same instrument. In our approach, we used a simple heuristic procedure to unify all the instances of the same instrument into a single sequence representing that instrument. The resulting sequence contains the figures according to the order given by the initial time of each figure, as shown in the example presented in Fig. 3.

Chords flattening At this point, we have a sequence of figures for each musical instrument. The last step in the transformation of the XML into sequences of figures consists in the elimination of chords that, from the point of view of rhythm, will be represented as a unique musical figure. To this aim, the sequences generated in the previous step are revised to detect musical figures with the same start and end times, keeping only one of them (Fig. 4).

Table 3 Computation of figures and rests

Lower range	Upper Range	Figure
T*2.5	infinite	Whole note
T*1.5	T*2.5	Half note
T*0.5	T*1.5	Quarter note
T*0.25	T*0.5	Eighth note
T*0.125	T*0.25	Sixteenth note
T*0.0625	T*0.125	Thirty-second note
0	T*0.0625	Sixty-fourth note

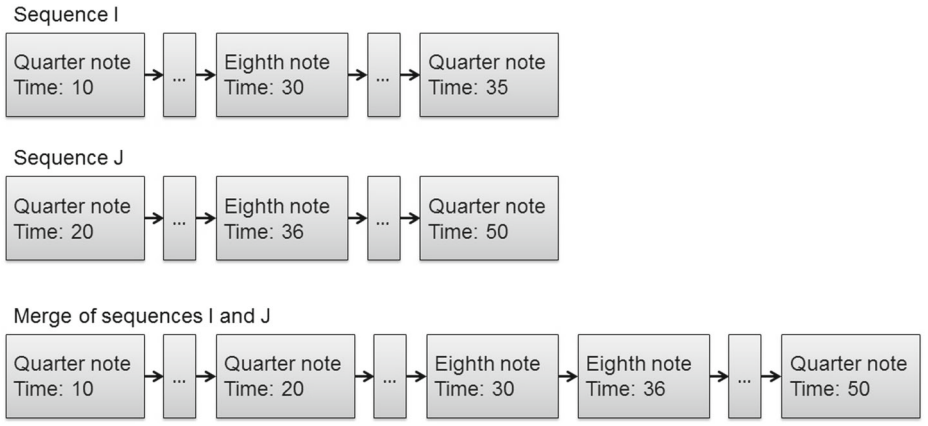

Fig. 3 Example of merging of two different sequences corresponding to the same instrument

5.1.2 Modelling sequences of figures

A natural way of modelling sequences of events observed along time is by using Markov models. In its simplest form, a Markov chain is a stochastic process with the Markov property. Having the Markov property means that, given the present state, future states are independent of the past states. In other words, the description of the present state fully captures all the information that could influence the future evolution of the process. Future states will be reached through a probabilistic process instead of a deterministic one. At each step the system may change its state from the current state to another state, or remain in the same state, according to a certain probability distribution. The changes of state are called transitions, and the probabilities associated with various state-changes are called transition probabilities. Markov chains of order N are a natural extension in which the future state is dependent on the previous N states. In many domains, models that consider longer states are beneficial obtain better performance. In our approach, we evaluated the performance of models of order 1, 2 and 3.

So the last step in the Data Transformation module consists in, given an order N, building a transition matrix with segments of size N taken from a sliding window moving along the sequences of figures. For example, for the sequence of figures C = {Quarter, Quarter, Half, Quarter rest, Eighth, . . . }, the first symbol of order 2 is A1={Quarter, Quarter} and the second is A2={Quarter, Half}.

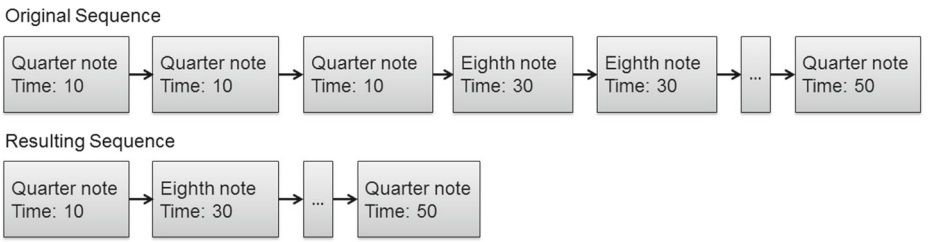

Fig. 4 Example of chords flattening

After observing the previous sequence of symbols, since there is a transition from A1 to A2, we sum 1 to the cell of the transition matrix corresponding to row A1 and column A2. Additionally, we build a vector with an entry for each symbol for counting the number of times each symbol was observed. When the process finalizes, we divide the number of occurrences of each symbol by the total number of observed symbols, obtaining a vector with the probabilities of occurrence of each symbol.

5.2 Classification

In this Section we describe the different classifiers implemented that are then combined with a voting scheme to predict the genre of a piece of music.

5.2.1 Transition matrix distance

With the transition matrix of figures for each instrument j playing in the piece of music, we can compute the Euclidean distance from this matrix to the matrix of the corresponding instrument for each genre in the knowledge base (1).

$$\Delta E_j = \frac{\sum_i (PA_i - PG_i)^2 \sum_k [(A_{ik} - G_{ik})^2 + (A_{ki} - G_{ki})^2]}{14} \tag{1}$$

In (1), A_i represents the transition matrix of figures for instrument i for the piece of music we want to classify and G_i represents the transition matrix of figures for instrument i for a given genre in the knowledge base. PA_i and PG_i represent the probability vector of symbols corresponding to matrix A_i and G_i respectively. The constant 14 is a normalization factor representing the worst case of the numerator (14 symbols in the matrix, 7 different figures and 7 different rests). The total Euclidean distance for a piece of music is computed by aggregating the Euclidean distance of each instrument and dividing by the probability of occurrence of the given instrument (2).

$$\Delta E = \frac{\sum_j \Delta E_j}{P_j} \tag{2}$$

For instruments that are present in the piece of music and not in the model of a given genre, we use a fixed threshold with the intention of incrementing the distance. In Section 6 we evaluate the impact of this threshold in the classification performance.

5.2.2 Figures prevalence

The figures prevalence classifier is based on the computation of n vectors $FP_A = FP_1, FP_2, \ldots FP_n$ corresponding to the occurrence probability of each figure for the n instruments present in the piece of music to be classified. In the same way, for each genre G in the knowledge base we will have k vectors $FP_G = FP_1, FP_2, \ldots FP_k$ containing the probabilities values of each figure for the k instruments observed for the genre. With this information, the Euclidean distance between each vector is computed (instrument by instrument), as shown in (3).

$$\Delta FP_x = \frac{\sum_{figure_i} (FP_{A_i} - FP_{G_i})^2}{IP_x} \tag{3}$$

Where x is a given instrument, and IP_x if the probability for instrument x in the model. The final distance between a musical piece and a genre in the knowledge base is given

by (4), where 128 is the total number of different instruments supported by the MIDI standard.

$$\Delta FP = \frac{\sum_{instrument_x} \Delta FP_x}{128} \tag{4}$$

5.2.3 Notes prevalence

The notes prevalence classifier is based on the computation of n vectors $NP_A = NP_1, NP_2, \ldots NP_n$ corresponding to the occurrence probability of each note for the n instruments present in the piece of music to be classified. Similarly, for each genre G in the knowledge base we will have k vectors $NP_G = NP_1, NP_2, \ldots NP_k$ containing the probabilities values of each note for the k instruments observed for the genre. With this information, the Euclidean distance between each vector is computed (instrument by instrument), as shown in (5).

$$\Delta NP_x = \frac{\sum_{note_i} (NP_{A_i} - NP_{G_i})^2}{IP_x} \tag{5}$$

Where x is a given instrument, and IP_x if the probability for instrument x in the model. The final distance between a musical piece and a genre in the knowledge base is given by (6), where 128 is the total number of different instruments supported by the MIDI standard.

$$\Delta NP = \frac{\sum_{instrument_x} \Delta FP_x}{128} \tag{6}$$

5.2.4 Instrumentation distance

Lets $I = I_1, I_2, \ldots, I_n$ be the collection of n instruments playing in a piece of music to be classified and $IO = IO_1, IO_2, \ldots, IO_k$ a vector counting the number of occurrences of the k instruments for a genre in the knowledge base. Then, for each instrument I_i, the occurrences are counted obtaining an indicator DI_g for each genre in the knowledge base. The lowest DI_g is selected as the candidate genre. For example, consider a piece of music A with the following instrumentation $IA = Piano, Acousticbass, Jazzguitar, Drums$ and assume that there are two genres in the knowledge base G_1 and G_2 with the following vector of occurrences of instruments: $IO_{G_1} = 10; 5; 8; 10$ and $IO_{G_2} = 10; 0; 10; 10$. Then, the total number of instruments occurrences for G_1 is 33 and for G_2 is 30, determining that the candidate genre for the piece of music A is G_1.

5.3 Coordination

The coordination component of Fig. 1, takes as input the different genres suggested by each classifier presented in Section 5.2 for a given piece of music and determines the final genre using a voting scheme. For weighting this voting scheme, the knowledge base keeps a confidence level for each classifier, which corresponds to the success rate obtained in the testing phase of the approach. If two or more classifiers assign the same genre to the piece of music, their confidence levels are averaged to compute the confidence in the genre.

For example, consider that the transition matrix distance classifier assigns the genre "bebop" to a given piece of music, the notes prevalence and figures prevalence classifiers assigns the genre "swing" to the same piece of music and the instrumental distance classifier assigns the genre "jazz soul". Then, if the confidence levels for each classifier are 0.90, 0.75, 0.80 and 0.20, the coordination module will assign a probability value of 0.90 to the

genre bebop, 0.77 to the genre swing and 0.2 to the genre jazz soul, assigning the genre bebop as the final genre of the piece of music.

5.4 User feedback

The proposed approach supports the user's feedback to adapt the classification models when new pieces of music are classified (either correctly or incorrectly). The adaptation process consists in averaging the data structures computed for the new piece of music with the data structures stored in the knowledge base.

By using the user feedback, each piece of music or set of pieces will enhance the classification models, reducing the effect of each transition between figures, each note and each instrument that is not typical for a given genre.

6 Experiments

In this Section we present and describe the classification results obtained by our approach and compare them with a McKay (2004), a competing approach. Bodhidharma uses a set of 100 features, a combination of neural networks with k-nearest neighbors to perform classification, and genetic algorithms for features selection and weighting.

McKay (2004) proposed a taxonomy consisting of 3 root genres and 9 sub-genres (leafs), as shown in Fig. 5. This taxonomy includes both categories with some similarity and categories that are considerably different.

The dataset used for this experiment consisted of 225 musical pieces, uniformly distributed according to the leaves in taxonomy presented in Fig. 5 (25 pieces for each leaf). Table 4 shows the average success rate at leaf level, for different threshold values and Table 5 shows the average success rate at root level. We can see that the success rate improves for lower threshold values, converging at 10^{-5}.

Using the same dataset, we established a threshold of 10^{-5} and performed a 5-fold cross validation, keeping the category distribution uniform (20 musical pieces corresponding to each subgenre were used for training and the remaining 5 pieces for testing. Table 6 shows

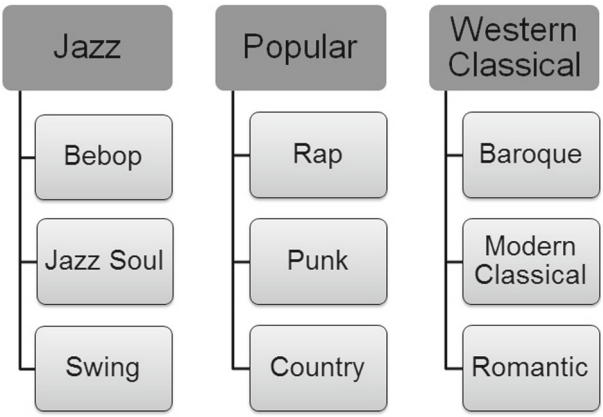

Fig. 5 Taxonomy used for classification

Table 4 Success rate at leaf level

Classifier / Threshold	1	10^{-3}	10^{-5}	10^{-8}
Transitions distance, order 1	0.289	0.600	0.556	0.556
Transitions distance, order 2	0.178	0.511	0.556	0.556
Transitions distance, order 3	0.133	0.311	0.556	0.556
Figures prevalence	0.244	0.556	0.556	0.556
Notes prevalence	0.156	0.489	0.489	0.489

the confusion matrix for the five folds average. Correctly classified instances are shown in bold.

The accuracy obtained by our approach was 0.8756, with 95 % CI in (0.8252, 0.9157), p-value $<$ 2.2e-16 and a Kappa coefficient of 0.86. We can infer from Table 6 that our approach was able to correctly identify all the songs belonging to the Swing sub-genre, within the Jazz genre. This is because its rhythm and instrumentation are very characteristic with the drums, the piano, the saxophone, the trombone and the bass almost always present. The same happens with the Punk sub-genre and the Traditional Country, under the Popular genre since these sub-genres have also a very singular rhythm and instrumentation, with a distinction of the drums, distorted guitar and bass. Similarly, the presence of the harpsichord, viola and violin for songs under the Baroque sub-genre, make it possible to discriminated this subgenre from other subgenres within the dataset.

We also test the same dataset with Bodhidharma tool. Average results for 5-fold cross validation are shown on Table 7. The accuracy obtained was 0.8267, with 95 % CI in (0.7708, 0.8737), p-value$<$2.2e-16 and a Kappa coefficient of 0.805. The general results of our approach are better than those obtained with Bodhidharma approach, which is very promising considering that Bodhidharma uses over 100 features, while we use only 3 features. We can observe that the genres that were better classified by this tool are the same than those better classified by our approach.

In order to visualize the impact of the number of features in the success rate, we performed a further experiment, selecting only 9 features from the 100 available. These features were selected according to their relationship with the features used by our approach: instrumentation, rhythm and pitch.

– For the instrumental aspect, we selected the following Bodhidharma features: Pitched instruments present (a boolean vector indicating the presence of absence of each instrument, except for percussion instruments), Unpitched instruments present (a boolean vector indicating the presence of absence of each percussion instrument), Number of pitched instruments and number of unpitched instruments

Table 5 Success rate at root level

Classifier / Threshold	1	10^{-3}	10^{-5}	10^{-8}
Transitions distance, order 1	0.553	0.778	0.756	0.733
Transitions distance, order 2	0.356	0.778	0.889	0.867
Transitions distance, order 3	0.333	0.533	0.533	0.722
Figures prevalence	0.511	0.778	0.778	0.778
Notes prevalence	0.333	0.756	0.756	0.756

Table 6 Performance of our approach (5-fold cross validation)

	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)
Beebop (A)	**0.80**	0.08	0.08	0	0	0.04	0	0	0
Jazz Soul (B)	0/08	**0.72**	0.16	0	0	0	0	0	0.04
Swing (C)	0	0	**1**	0	0	0	0	0	0
Hardcore rap (D)	0.04	0.04	0	**0.88**	0	0.04	0	0	0
Punk (E)	0	0	0	0	**1**	0	0	0	0
Traditional Country (F)	0	0	0	0	0	**1**	0	0	0
Baroque (G)	0	0	0	0	0	0	**0.96**	0	0.04
Modern Classic (H)	0	0	0.8	0	0	0	0	**0.64**	0.28
Romantic (I)	0	0.04	0	0	0	0	0.04	0.04	**0.88**
Precision	**0.87**	**0.82**	**0.76**	**1**	**1**	**0.93**	**0.96**	**0.94**	**0.71**
Recall	**0.80**	**0.72**	**1**	**0.88**	**1**	**1**	**0.96**	**0.64**	**0.88**

– For the tonal aspect, we selected the following Bodhidharma features: Note prevalence of pitched instruments (for each note, this feature count the total number of NoteOn events), Note prevalence of unpitched instruments (the same as before, but for percussion instruments), Variability of note prevalence of pitched instruments (standard deviation for each note performed by each instrument), and Variability of note prevalence of unpitched instruments (the same as before, but for percussion instruments).
– To cover the rhythm aspect, we selected the Time prevalence of pitched instrument, since it is based on the occurrence of different musical figures. This feature encodes the percentage of the total time of the music piece in which each different musical figure appears.

Table 7 Performance of Bodhidharma approach for all features (5-fold cross validation)

	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)
Beebop (A)	**0.80**	0.16	0.04	0	0	0	0	0	0
Jazz Soul (B)	0.20	**0.60**	0.08	0.04	0	0.04	0	0.04	0
Swing (C)	0	0.04	**0.96**	0	0	0	0	0	0
Hardcore rap (D)	0	0.12	0	**0.80**	0	0.08	0	0	0
Punk (E)	0	0	0	0	**1**	0	0	0	0
Traditional Country (F)	0	0.04	0	0	0	**0.96**	0	0	0
Baroque (G)	0	0	0	0	0	0	**0.96**	0	0.04
Modern Classic (H)	0	0.08	0	0	0	0	0	**0.52**	0.40
Romantic (I)	0	0	0	0	0	0.04	0	0.12	**0.84**
Precision	**0.80**	**0.58**	**0.89**	**0.95**	**1**	**0.86**	**1**	**0.76**	**0.65**
Recall	**0.80**	**0.60**	**0.96**	**0.80**	**1**	**0.96**	**0.96**	**0.52**	**0.84**

Table 8 Performance of Bodhidharma approach with 9 manually selected features (5-fold cross validation)

	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)
Beebop (A)	**0.68**	0.28	0.04	0	0	0	0	0	0
Jazz Soul (B)	0.24	**0.47**	0.05	0.10	0.02	0.08	0	0.04	0
Swing (C)	0	0.06	**0.94**	0	0	0	0	0	0
Hardcore rap (D)	0	0.12	0	**0.80**	0	0.04	0	0.04	0
Punk (E)	0	0	0	0	**1**	0	0	0	0
Traditional Country (F)	0	0.04	0	0	0	**0.96**	0	0	0
Baroque (G)	0	0	0	0	0	0	**0.92**	0	0.08
Modern Classic (H)	0	0	0	0	0	0	0.08	**0.44**	0.48
Romantic (I)	0	0	0	0	0	0	0.08	0.32	**0.60**
Precision	**0.74**	**0.48**	**0.91**	**0.88**	**0.98**	**0.86**	**1**	**0.76**	**0.52**
Recall	**0.68**	**0.47**	**0.94**	**0.80**	**1**	**0.96**	**0.96**	**0.52**	**0.60**

The results obtained with this configuration are shown in Table 8. As we can see, the performance was better for our approach than for Bodhidharma.

Finally, we use the feature selection capability of Bodhidharma to automatically select from all the available features, those features that better describe the training examples without losing any important information. Bodhidharma uses genetic algorithms with a roulette crossover system with mutation and with elitishm but without villages (McKay 2004). Table 9 shows the results obtained for this experiment. The accuracy obtained with this configuration was 0.8311, 95 % CI in (0.7756, 0.8776), p-value<2.2e-16 and a Kappa coefficient of 0.81. We can observe that performance of Bodhidharma with features selections is better than without features selection, but it still under the performance of the approach presented in this article.

Table 9 Performance of Bodhidharma approach with feature selection enabled (5-fold cross validation)

	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)
Beebop (A)	**0.76**	0.2	0.04	0	0	0	0	0	0
Jazz Soul (B)	0.2	**0.72**	0.04	0	0	0	0	0.04	0
Swing (C)	0	0.04	**0.96**	0	0	0	0	0	0
Hardcore rap (D)	0	0.08	0	**0.88**	0	0	0.04	0	0
Punk (E)	0	0	0	0	**1**	0	0	0	0
Traditional Country (F)	0	0.08	0	0	0	**0.92**	0	0	0
Baroque (G)	0	0	0	0	0	0	**0.96**	0	0.04
Modern Classic (H)	0	0	0	0	0	0	0	**0.56**	0.44
Romantic (I)	0	0.04	0	0	0	0	0.04	0.2	**0.72**
Precision	**0.79**	**0.62**	**0.92**	**1**	**1**	**1**	**0.92**	**0.74**	**0.58**
Recall	**0.76**	**0.72**	**0.96**	**0.88**	**1**	**0.92**	**0.96**	**0.56**	**0.72**

7 Discussion and conclusions

In this article, we proposed an approach for efficient genre classification of symbolic musical pieces using only three features. We conducted an experimental evaluation using a taxonomy consisting in three root genres and nine subgenres and compared the results obtained with those presented by Bodhidharma, a related approach proposed by McKay (2004). Although McKay approach used 100 music features to perform classification, while our approach uses only 3, we obtain better classification performance.

Our approache obtained an average accuracy of 87.56 % correctly classified songs, while with Bodhidharma approach we obtained, for the same dataset, 82.67 % correctly classified instances (a significant improvement of 4.89 %, $p<0.05$, $t=3.0509$). In order to make the number of features more comparable for both approaches, we conducted a second experiment using only the nine features more related to the features used in our approach. The accuracy of Bodhidharma in this second experiment was 75.7 %. In conclusion, our approach demonstrated to have also a better accuracy, with a significant improvement of 11.89 % ($p<0.05$, $t=3.5383$). Finally, if we let Bodhidharma to automatically select the features that better describe the training data, it obtained a better performance (83.11 % accuracy) than that obtained using the complete set of features. However, our approach still have a significant improvement of 4.45 % ($p<0.05$, $t=2.6786$).

Genres in which our approach performed worst are mainly post-baroque western classical music (Romantic and Modern classical). These genres correspond to music mainly composed between 1780 and 1975 and are characterized by changing harmonies conducting to modulations to distant and usually unexpected tones. This kind of music is also characterized by the occurrence of pitches in novel combinations, as well as by the occurrence of familiar pitch combinations in unfamiliar environments. As shown in Table 6, our approach was not able to distinguish between these two genres, and often classified instances of Modern classical as Romantic music.

If we consider the performance at root level, that is if the classifier was able to predict the top level genre (Jazz, Popular, Western classical), we obtained an average accuracy of 95.1 %., while bodhidharma tool we obtained an average of 95.6 % using all features and 95.1 % using 9 features and 97.1 % using feature selection. In all cases, the difference in not statiscally significant ($p=0.87$, $p=1.00$ and $p=0.64$, respectivelly) and then we do not have enough information to make any conclusions for the classification performance at root level.

Several authors (Lippens et al. 2004; Gjerdingen and Perrott 2008; Seyerlehner 2010) have shown that human agreement on which music pieces belong to a particular genre ranges only between 55 % and 76 %. Lippens et al. (2004) performed a listening experiment with 27 humans in which they were asked to classify 160 songs into six possible genres by listening to 30 seconds of audio, obtaining an average performance of 76 %. Gjerdingen and Perrott (2008), performed a similar experiment in which a set of students were asked to identify the genre (blues, country, western classical, dance, jazz, latino, pop, R&B, rap y rock) of different musical pieces by listening three seconds of audio, the success rate was 70 %. On the other hand, Seyerlehner (2010) performed the listening experiment with 24 humans and 19 genres. Participants were able to listen to the complete songs to be classified, obtaining an average classification accuracy of 55 %. Although these results cannot be straightforwardly compared with the results reported in this article, they give an idea of the baseline that can be considered a good classification result.

An important implication of our results is the demonstration of the potential power of high-level features in music classification. An interesting advantage of our approach with regards to Bodhidharma is that, we consider the user feedback to update the knowledge

base after each classification. This way, it is expected that our approach can improve the classification performance when more examples of each genres are provided for training the different classifiers.

References

Abeßer, J., Lukashevich, H., & Bräuer, P. (2012). Classification of music genres based on repetitive basslines. *Journal of New Music Research*, *41*(3), 239–257.

Aucouturier, J.-J., & Pachet, F. (2003). Representing musical genre: A state of the art. *Journal of New Music Research*, *32*(1), 83–93.

Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., & Klapuri, A. (2013). Automatic music transcription: Challenges and future directions. *Journal of Intelligent Information Systems*, *41*(3), 407–434. doi:10.1007/s10844-013-0258-3. ISSN 1573-7675.

Chai, W., & Vercoe, B. (2001). Folk music classification using hidden markov models. In *Proceedings of international conference on artificial intelligence*.

Chen, G.-F., & Sheu, J.-S. (2014). An optical music recognition system for traditional chinese kunqu opera scores written in gong-che notation. *EURASIP Journal on Audio, Speech, and Music Processing*, *7*, 1–12.

Dannenberg, R. B., Thom, B., & Watson, D. (1997). A machine learning approach to musical style recognition. In *Proceedings of the international computer music conference* (pp. 344–347).

Downie, J.S. (2003). Music information retrieval. In: *Annual review of information science and technology* (pp. 37: –).

Fabbri, F. (1999). Browsing music spaces: Categories and the musical mind. In *Proceedings of the IASPM-US conference (International Association for the Study of Popular Music)*.

Gjerdingen, RO., & Perrott, D. (2008). Scanning the dial: The rapid recognition of music genres. *Journal of New Music Research*, *37*(2), 93–100.

de Jesus Guerrero-Turrubiates, J., Gonzalez-Reyna, S. E., Ledesma-Orozco, S. E., & Avina-Cervantes, J. G. (2014). Pitch estimation for musical note recognition using artificial neural networks. In *Proceedings of the 24th international conference on electronics, communications and computers* (pp. 53–58).

Kotsifakos, A., Kotsifakos, E. E., Papapetrou, P., & Athitsos, V. (2013). Classification of symbolic music with smbgt. In *Proceedings of the 6th international conference on PErvasive technologies related to assistive environments*.

Lippens, S., Martens, J.P., De Mulder, T., & Tzanetakis, G. (2004). A comparison of human and automatic musical genre classification. In *Proceedings of the IEEE international conference on acoustics, speech, and signal processing* (vol. 4).

McKay, C. (2004). Automatic genre classification of midi recordings. Master's thesis, McGill University, Canada.

Ponce de León, P. J., & Iñesta, J.M. (2002). Musical style identification using self-organising maps. In: *Proceedings of the second international conference on web delivering of music* (pp. 82–89).

Schedl, M., Gómez, E., & Urbano, J. (2014a). Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, *8*(2–3), 127–261.

Schedl, M., Hauger, D., & Urbano, J. (2014b). Harvesting microblogs for contextual music similarity estimation: a co-occurrence-based framework. *Multimedia Systems*, *20*(6), 693–705.

Seyerlehner, K. (2010). Content-Based Music Recommender Systems: Beyond simple Frame-Level Audio Similarity. PhD thesis, Johannes Kepler University Linz, Linz, Austria.

Shan, M.-K., Kuo, F.-F., & Chen, M.-F. (2002). Music style mining and classification by melody. In *Proceedings of 2002 IEEE international conference on multimedia and expo* (pp. 97-100).

Li, S., & Yang, Y.-H. (2015). Combining spectral and temporal representations for multipitch estimation of polyphonic music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(10), 1600–1612.

Tagg, P. (1982). Analysing popular music: theory, method and practice. *Popular Music*, *2*, 37–67.

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

Valverde-Rebaza, J., Soriano, A., Berton, L.n., Oliveira, M.C.F., & de Andrade Lopes, A. (2014). Music genre classification using traditional and relational approaches. In *Proceedings of the Brazilian conference on intelligent systems*.

Wen, C., Rebelob, A., Zhang, J., & Cardosob, J. (2015). A new optical music recognition system based on combined neural network. *Pattern Recognition Letters*, *58*, 1–7.

Wieczorkowska, A.A., & Ras, Z.W. (2003). Editorial—music information retrieval. *Journal of Intelligent Information Systems*, *21*(1), 5–8. doi:10.1023/A:1023543016136. ISSN 1573-7675.

Wojcik, J., & Kostek, B. (2010). *Representations of music in ranking rhythmic hypotheses, chapter 3*, (pp. 39–64). Berlin Heidelberg: Springer. ISBN 978-3-642-11674-2.