Mining in Music Databases

INTRODUCTION

Musical analysis is recognised as a significant part of the study of musical cognition. The analysis of music data has the objective of determining the fundamental point of contact between mind and musical sound (musical perception) (Bent, I., 1980). Musical analysis is the activity musicologists are engaged to and is conducted on a single piece of music, on a portion or element of a piece or on a collection of pieces. This research area embays the field of *music data mining* (henceforth called *music mining*), which deals with the theory and methods of discovering knowledge from music pieces and can be considered as a collection of (semi-) automated methods for analysing music data.

Following music-mining methodologies, music analysts extract[1] recurring structures and their organisation in music pieces, trying to understand the style and techniques of composers (Rolland, P.Y. & Ganascia, J.-G., 2002). However, the size and peculiarities of music data may become prohibitive factors for the aforementioned task. This represents an analogy to the difficulties faced by data analysts when trying to discover patterns from databases, i.e., the huge database sizes and the large number of dimensions, which are the very reasons that paved the way to the development of *database mining* a.k.a. *data mining or knowledge discovery from databases* (*KDD*). Despite the previously mentioned analogy between music mining and database mining, the nature of music data requires the development of radically different approaches. In the sequel of this section we will summarise the particular challenges that music mining presents.

Another key issue in which music mining differs from other related areas (for instance, database mining or web mining) is the applications it finds. Discovered patterns from relational or other types of databases, are usually *actionable*, in the sense that they may suggest an action to be taken. For instance, association rules from market-basket data can indicate an improvement in selling policy, or user-access patterns extracted from a web-log file can help in redesigning the web site. Such kind of "actionability" is related to a form of "profit" and stems from the involved industry field (e.g., retail, insurance, telecommunications, etc). The question, therefore, emerges: "Which is the usability of patterns extracted from music data?". In order to answer this question, one has to consider the current status of the involved industry, that is, the "music industry". The influence that music always had on people is reflected in music commodities and services that are offered today.[2] The annual gains of the music industry are estimated to reach up to several billions dollars (Leman, M. 2002). Within this context, the music content is a source of economical activity. This is intensified by the ease that the Web has brought in the delivery of music content; a prominent example of this case is Napster. What is, thus, becoming of significant interest is the need for content-based searching within music collections, e.g., by using a Karaoke machine to retrieve similar songs over a web site or by humming over a mobile phone to download a song. The corresponding research field that has been developed is called *content-based music information retrieval* (CBMIR).

It is natural, therefore, to anticipate that music mining finds applications in designing effective CBMIR systems. In fact, CBMIR has considerably biased the directions that research in music mining is now following, by stating the objectives to be achieved. The contribution of music mining in CBMIR is better understood by considering that the extracted patterns describe and represent music content at different abstraction levels (e.g., by producing concept taxonomies). The description of music content with such representations helps users in posing queries using content descriptors (rational or emotional), which drastically improve the effectiveness of retrieval in CBMIR systems (Leman, M. 2002), compared to simplistic search using plain text descriptors like song tiles or the composers' names. Additionally, searching times are decreased, since the extracted patterns constitute a more compact representation of

[1] (Rolland, P.Y. & Ganascia, J.-G., 2002) makes an interesting distinction between *pattern discovery* and *pattern extraction*. The former refers to the detection of local regularities in data. The latter also refers to such kind of detection, but is additionally concerned with the explicit availability of the patterns in some language, at the end of the mining process.

[2] To name just few: music sold as CDs, concerts, broadcasting of video-clips in mass media, advertised products related to music and music performers, and, more recently, on-line sales of music in electronic format.

music content. The advantages from both the aforementioned directions are evident in a broad range of commercial domains, from music libraries to consumer oriented e-commerce of music (Rolland, P.Y. & Ganascia, J.-G., 2002).

The Challenges of Music Data Mining

Byrd and Crawford (Byrd D. & Crawford T., 2002) list a number of reasons for which it is difficult to manage music data. Some of them also affect the case of music mining. The most basic difficulty stems from the fact that it has not been clear so far how to automatically segment music data into meaningful units, like music *phrases* or *motives*. In order to be able to extract patterns, it is reasonable to desire the representation of music in such units, in the way that text mining considers words as units of text data that are used pattern searching[3]. Although the difficulty is larger for acoustic representations of music, it is not easy to detect basic units in symbolic representation as well (for representation types, see the following section on music databases). The problem is further intensified by considering that there exists an amount of overlapping between music units; one such case is due to polyphony.

Polyphony is the source of more difficulties. For simplicity, initial attempts in music mining focused on symbolic representation of monophonic music. Polyphony, however, is present in almost all real works of music. The difficulty is rising from the fact that it is required to separate simultaneous independent voices in order to distinguish them, in the same way that one separately recognises the lines of each character in a play. This problem is considered as the most intractable and can significantly impact the quality of the analysis (Byrd D. & Crawford T., 2002), as music phrases may appear audibly few times although they may occur frequently in the music score (e.g., buried within repeated chords).

Repetition of occurrence signals a pattern in almost all mining fields (database, web, or text mining). However, in all these fields, patterns are repeated *exactly*. In music mining, one should not focus on exact repetition of patterns, due to *variation* and *ornamentation* that are present in music data (Rolland, P.Y. & Ganascia, J.-G., 2002). Therefore, algorithms that search for music patterns should take into account this peculiarity as well.

Data quality is a factor that is taken into account by all mining fields. Data cleansing methods are used to avoid the discovery of patterns that will lead to pitfalls. Therefore it comes at no surprise that music data are very prone to errors, since there is very little quality control of publicly available music data (Byrd D. & Crawford T., 2002). However, music data have an additional source of “error”, which is the result of differences in performance-related characteristics; differences in key, tempo, or style cause different instantiations of an identical musical score. These factors have to be additionally taken into account when searching for patterns.

All the aforementioned issues concern the effectiveness of the music mining process. Another important aspect is efficiency. Music databases tend to be large, both due to the large number of music pieces they contain and the large size of each piece. The challenge is, therefore, to develop scalable algorithms for music mining. It is worth noticing that many existing approaches are influenced from *soft-computing* methodologies (e.g., neural networks, SVMs, genetic algorithms), which do not scale very well. Moreover, algorithms for finding repeating patterns are confined to main-memory resident data. Hence, another challenge is to develop algorithms for disk-resident data.

Finally, it must be argued that although the incorporation of background knowledge in the mining process is considered important, it is generally a vague issue. It seems that domain-specific knowledge in music mining is sine-qua-non. What is, therefore, required is the systematic development of methods to incorporate this knowledge in the music mining process, a task that is very hard to consider.

Chapter outline

In what follows this chapter, we summarise existing work on music mining. First, we give the necessary background on music databases. Next, we examine the task of similarity searching, which has attracted significant attention in research related to CBMIR. Similarity searching in music mining is of high importance, as it serves as a primitive for more complex ones. In the two sections that follow, we study methods for clustering and classification of music data. Clustering methods are for unsupervised learning, whereas supervised learning methods have been mainly used for tasks such as genre classification. Next, we

[3] Notice that when mining from relational or other highly structured data, this problem is not present at all, since units of information are well defined by the schema of the database.

move on to examine algorithms for detecting repeating patterns, and we also discuss the special issue of theme finding. In the final section, we conclude this chapter and present the perspective of music mining.

MUSIC DATABASES

It is only the last decade that large scale computer-based storage of sound and music has been possible. Additionally, the increasing ease of distribution of music in computer files over the Internet gave further impulse to the development of digitised music databases as well as to new methods for Music Information Retrieval (MIR) in these collections.

Initial efforts for Information Retrieval (IR) in music databases relied on the well-studied text IR, that is, on the metadata of the music objects (title, composer, performer, genre, date, etc. – see the extension of the mp3 format, called ID3-Tag, for an example). Although abundantly used, even nowadays, the traditional metadata of a music object give rather minimal information about the actual content of the music object itself. Moreover, metadata in most cases are manually maintained, therefore this process is notoriously time consuming. On the other hand, queries based on humming (using a microphone) or on a small piece of musical file, are a more natural approach to MIR. This type of queries lies within the CBMIR. In CBMIR, an actual music piece is required in order to compare its content with the content of the music pieces already available in the database.

Music Data Representation and Format Conversion

Music is available in two basic representations: the symbolic representation (MIDI, Humdrum format) and the acoustic representation (audio format - wav, mp3, wma, etc). Their key difference lies in the fact that the family of symbolic representations carries in their objects information of what a musical player should perform, whereas the acoustic representations comprise a specific recorded performance of a music piece. In other words the term music encompasses both performance directions as well as resulting sounds.

The symbolic representation can further be separated into two classes according to the targeted performer. Thus, there exist symbolic representations aimed at digital devices such as the MIDI and Humdrum formats as well as human oriented symbolic representations that are collectively referred to as Conventional Music Notation (CMN).

A MIDI (MIDI is the Musical Instrument Digital Interface specification) object consists of predefined “events” that are quantified factors that define a musical performance (Owen, S. R., 2000). Typical such events include the notes to be played, the time instance and the force these notes should be played, the type of instrument playing them, just to name a few. Following the high detail an event may contain, a MIDI object can quite accurately describe a music performance and thus its use is rather popular, especially for classical music playback. Additionally, the MIDI format is also used in order to communicate music between digital devices since it is codified[4] and has wide acceptance thus offering interoperability between different types of music-aware devices.

CMN commonly includes numerous features that are not defined in the MIDI protocol, such as rests, slurs, barlines, triplets and chromatisms. O’Maidin et al. (Maidín, D.Ó & Cahill, M., 2001) propose a complex object framework that serves as a container of a collection of objects modelled for music scores as well as iterators for use with the algorithms available, the C.P.N.View (Common Practice Notation View). C.P.N.View is a class library for representing musical scores in a form suitable for arbitrary processing. Another approach for a score-based music representation is presented by Hoos et al. in (Hoos, H.H., Hamel, K.A., Renz, K., & Kilian, J., 1998), which utilises the GUIDO Music notation. The GUIDO Music Notation Format is a novel, general purpose formal language for representing score level music in human-readable way.

Music in the acoustic representation consists of a time-series of sampled signal amplitude values. These values represent the air pressure changes created by the music source that propagate from the source (e.g. loudspeaker or violin) to the listeners ears as air pressure fluctuations. A very simple method of illustration of acoustical signals is by drawing these signals as a graph of air pressure versus time

[4] See: “The Complete Detailed MIDI 1.0 Specification”, MIDI Manufactures Association, 1996.

(figure 2.a.1). In that case, the representation of the acoustical signal is called time-domain representation and the amplitude of the waveform is the amount of air pressure change (Roads, C., 1996)

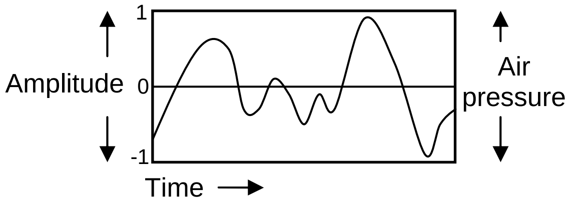

Figure 2.a.1: Time domain representation of a signal.

Since the conversion of symbolic music to and from CMN is generally admitted to be easy, in this section the interest is brought on the process of analysing music in the acoustic representation so as to identify the elements that constitute the piece of music in question (Klapuri, A., 2004). This process is known as music transcription. The notation utilised for the symbolic representation of the acoustic format can be any symbolic representation offering sufficient information for the transcribed piece to be performed.

Although skilled musicians are able to perform music transcription with high success (Klapuri, A., 2004), computer music transcription is generally admitted to be very hard and poor performing (Yang, C., 2002; Pickens, J., 2004). The performance of the computer systems degrades even more when the transcribed music piece is polyphonic[5].

The last ten years of research in polyphonic music transcription brought a great deal of knowledge to the area, though no generally applicable all-purpose system exists. The latest proposals presented a certain degree of accuracy in limited complexity polyphonic music transcription (Davy, M. & Godsill, S. J., 2003; Bello, J. P., 2003). Their common limitations include as prerequisites, for acceptable performance, a specific number of concurrent sounds and absence of percussive instruments.

Despite the unfavourable research template, a small number of commercial approaches to a music transcription system have been available (AKoff, 2001; Arakisoftware 2003, Innovative 2004; Seventh, 2004][6], though their performance is rather poor as far as accuracy is concerned (Klapuri, A., 2004).

Music Features

Music consists of numerous features. Among them, pitch, rhythm, timbre, and dynamics are considered to be the most semantically important ones (Byrd D. & Crawford T., 2002). For western music in particular, pitch carries the highest relative weight of information followed by rhythm (Byrd D. & Crawford T., 2002).

For the MIR process to perform matching algorithms on the music data, descriptions of these features for the music data are necessary. Thus, the previously mentioned representations require a conversion from their original format to the format defined by each MIR system. The conversion process is also known as feature extraction. The selection of features to be conversed by the feature extraction process is implementation dependent. That is, a variety of alternatives exist with respect to the characteristic of music that should be included in the final format.

The feature selection and extraction process can be separated based on the representation of the music piece in symbolic and acoustic feature extraction, while the former can also be divided into monophonic, homophonic and polyphonic.

Music in the form of acoustic representation requires special analysis in order to extract features such as pitch and rhythm, while the non-triviality of the problem is reflected by the number of methods developed. The key idea in this case is audio transcription to feature events. The most common features (Wieczorkowska, A. & Ras, Z., 2001) are the coefficients of time-domain analysis (Papaodysseus, et al. 2001; Paraskevas, M. & Mourjopoulos, J., 1996), spectral analysis (Papaodysseus, et al; Paraskevas, M. & Mourjopoulos, J., 1996; Kostek, B. & Wieczorkowska, A., 1997) and wavelet analysis (Wieczorkowska, A., 2001).

[5] A polyphonic music piece, for the purposes of this work, refers to a music piece wherein at any discrete time instance more than one sounds may occur.
[6] This list is by no means exhaustive. It merely presents a few of the latest systems.

Pitch detection[7] is dealt with time-domain fundamental period pitch detection, autocorrelation pitch detection, adaptive filter pitch detection, cepstrum analysis and frequency-domain pitch detection. It should be noted that no pitch detection algorithm is totally accurate and some that appear to be, utilise music inputs that follow specific constraints or show increased computational requirements (non real-time). Key difficulties in pitch detection include attack transients, low and high frequencies identification, myopic pitch tracking and acoustical ambience.

In the case of polyphonic acoustic signals, the complexity rises additionally, while attempts towards this direction usually apply frequency-domain analysis techniques within a fundamental pitch or strong harmonics selective mechanism.

Rhythm detection can be divided into three levels: low-level (event detection), mid-level (transcription into notation) and high-level (style analysis). As with pitch detection, rhythm detection is also inherently difficult due to non accurate human performance of musical scores as well as the ambiguity of the music notation[8].

Moving on to the music in the form of symbolic representation, feature extraction from MIDI-like music files is rather easier as the results of the transcription the step of the acoustic representation feature extraction are, apparently, already available in some form.

In monophonic music no note may sound until the currently sounding has finished, in homophonic music, simultaneous notes may start and finish sounding together, while in polyphonic music a note may sound before a previous one finished. Addressing the homophonic and polyphonic music is achieved by reduction to monophonic and homophonic respectively and by modification of the methods utilised for the monophonic music.

The problem of monophonic music symbolic feature extraction can be reduced to *n*-dimension by retaining only *n* different information described in the music file. Additional approaches include *N*-grams (e.g. sliding windows, repeating patterns) and shallow structure methods (Pickens, J., 2001) (e.g. statistical measures, lightweight computation and music theory analysis).

Monophonic reduction is an initial attempt to solve the problem of simultaneous notes sounding by selection of only one of the notes sounding in any discrete time. Accordingly, monophonic methods can be utilised for feature extraction from the received monophonic music. The key issue in monophonic reduction is the “representative” note selection method.

In homophonic reduction, instead of selecting only one note at each time step, all notes at each time step are retained and the reduction that incurs is assuming independence of overlapping duration notes (Pickens, J., 2001).

Finally, as a means of features that can be indexed, clustered and matched for similarity research has used Hidden Markov Models (HMM). An HMM is a Markov chain, where each state generates an observation. One can only see the observations, and the goal is to infer the hidden state sequence. From the observable sequence of outputs, infer the most likely dynamical system. The result is a model for the underlying process. Alternatively, given a sequence of outputs, infer the most likely sequence of states.

Hidden Markov Models (HMM) have been extensively used in MIR. Numerous approaches (Pikrakis, A., Theodoridis, S., & Kamarotos, D., 2002); Shifrin, J., Pardo, B., Meek, C., & Birmingham, W., 2002; Velivelli, A., Zhai, C., & Huang, T.S., 2003] have utilised HMMs in order to represent music pieces in a database and the queries posed. In (Pikrakis, A., Theodoridis, S., & Kamarotos, D., 2002), Pikrakis et al. present a method for automated search of predefined sound patterns within a large number of sound files, using HMMs. In (Shifrin, J., Pardo, B., Meek, C., & Birmingham, W., 2002) the authors use a stochastic representation of both music sequences in the system and the queries, with hidden Markov models in order to handle queries that contain errors or key and tempo changes. Velivelli et al. (Velivelli, A., Zhai, C., & Huang, T.S., 2003) utilise HMMs that can model predefined patterns and simultaneously identify and match an audio segment for a given query.

Indices for Music Data

The selection of appropriate features is considered very important in multimedia information retrieval. Meaningful features not only help in the effective representation of the objects but also enable the use of indexing schemes for efficient query processing.

[7] For a broader analysis readers are referred to (Roads, C., 1996).
[8] For a broader analysis readers are referred to (Roads, C., 1996).

As far as the symbolic representation is concerned, recent research proposed that features could be represented in string format and accordingly presented string indices (Hsu, J., Liu, C., & Chen, A.L.P., 1998; Lee, W. & Chen, A.L.P., 2000; Chen, J.C.C. & Chen, A.L.P., 1998; Liu, C.C., Hsu, J.L., & Chen, A.L.P., 1999). Though, these approaches are not easily adopted for multiple features, lack data scalability for large music data (Lee, W. & Chen, A.L.P., 2000) and generally string matching proves slower than numeric matching (Lo, Y.L. & Chen, S.J., 2003). In order to address these inefficiencies, (Lo, Y.L. & Chen, S.J., 2003) proposed a multi-feature numeric indexing structure that transforms music feature strings into numeric values.

As already mentioned, the most common features of acoustic music are produced by time analysis, spectral analysis and wavelet analysis. The coefficients collected from each of these analyses can be indexed in TV-Trees (Lin, K.-I., Jagadish, H.V., & Faloutsos, C., 1994), locality-sensitive hashing schemes (Yang, C., 2002), S-Indexes (Dervos, D., Linardis, P., & Manolopoulos, Y., 1997). In addition, (Reiss, J., Aucouturier, J.-J., & Sandler, M., 2001) compares four different multidimensional indexing schemes for music data, the KD-Tree, the K-Tree, the Multidimensional Quick-sort and the Box Assisted Method. The authors conclude that KD-Tree is significantly more efficient than the other methods, especially for high-dimensional data. Finally, the authors in (Won, J.-Y., Lee, J.-H., Ku, K., Park, J., & Kim, Y.-S., 2004), utilise an M-Tree in which a selection of features is stored, claiming thus a 65% gain in space requirements.

SIMILARITY SEARCHING

As a result of the rapid developments in WWW, users are able to search for music information among various and large amounts of music data (this is especially popular within P2P systems). This reveals the need for development of effective similarity-searching methods for music data. Additionally, another significant area where similarity-searching is applied is CBMIR systems. Similarity searching in music data differs from searching other types of data in the following: (i) users may perceive in different ways the notion of similarity between music pieces, (ii) the possibility of ad-hoc nature in similarity-searching query posing (e.g., querying by humming), which brings the need for high tolerance against inconsistencies, (iii) the influence of data representation (symbolic or acoustic) on the designation of similarity-searching algorithms. The importance of similarity searching stems from its special role as primitive for other data mining tasks, like information retrieval, clustering and classification.

Perception of Similarity in Music Data

The stimuli received by a human observer lead to the experience of the event that produced these stimuli, through the interpretation process of the brain. The reception made through the five senses available to humans, is the sole contributing channel of information. Though, the final representation, of the event that produced the stimuli, in the human brain has little direct relevance with what the sensory transducers received, as it is subject to extended further processing by the brain (Shepard, R., 1999). Accordingly, cognitive psychology studies the final representation corresponding to the perceived stimuli.

Music, being a physical stimulus as well, is amenable to the very same extended brain processing after being received by the acoustical[9] system. Thus, musical cognition is up to a certain degree subjective (perception ambiguity) while numerous preferred or habitual ways of music cognition/listening do exist.

Based on the Gestalt Laws of Grouping, a number of principles exist by which people organise isolated parts of a visual or acoustic stimulus into groups or whole objects. There are five main laws of grouping: *proximity*, *similarity*, *continuity*, *closure*, and *common fate*. All of these laws fall under a sixth law, that of *simplicity*. Although Gestalt laws are usually applied to visual stimuli, their appliance in other senses, such as the auditory, is well known. According to the Gestalt laws, during the experience of music listening, humans do not hear a series of disconnected or random tones. Music is perceived as a whole by means of sound relation based pitch similarity, time proximity and other factors. Music perception can lead to the identification of melodies, patterns and forms.

---

[9] Acoustical pertains to the objective physics of a sound. For example, frequency is a physical or acoustical property, whereas pitch is a subjective or auditory property

Recent work by McAdams et al. (McAdams, S., Vieillard, S., Houix, O., & Reynolds, R., 2004) suggested that music similarity is determined based on surface features of the musical material. These findings are also supported by numerous previous works on the field. Additionally, the listener sensitivity to these features is not related to musical education, apart from the terminology used in order to express the similarity or difference. The work considers as surface features duration/rhythm, pitch/melody, timbre, gesture, texture, articulation and to a lesser degree dynamics and harmony, although, some of these features may be more or less overlapping.

Additionally, based on studies that have been conducted on what people attend to while detecting similarity, musicians initially observed dynamics, art, texture, and then noticed pitch height and contour, while non-musicians attention was firstly drawn to dynamics and art, while texture and pace were subsequently noticed (Lamont, A. & Dibben, N., 2001).

Musical recognition is greatly affected by the ability of the listener to detect different levels of similarity in different musical elements. Studies on response times in dissonant and consonant differentiation proved that distinguishing dissonant than consonant chords is faster, while musical training had a great effect in number of correct responses (Bigand, E., Madurell, F., Tillmann, B., & Pineau, M., 1999). However, in experimentation in chord similarity perception, experience did not seem to have an effect on perception (Hubbard, T.L., 1996).

The mood or emotional atmosphere evoked when listening to a musical piece was revealed, by McAdams et al. (McAdams, S., et al., 2004), to be relative to musical similarity.

In the perception of musical emotion, timbre, tempo, pitch and rhythm are of critical importance. Fast tempos are regarded as happy, joyful, while slow tempos tend to be considered as sad, gloomy (White, R.E., 2004). A more or less same effect appears with pitch, with high and simple pitches (melodies) producing a feeling of happiness and lower and more complex pitches (melodies) sadness. The combination of rhythm and pitch is known to affect the degree a musical piece is scarring (Schellenburg, E.G., Krysciak, A. M., & Campbell, R.J., 2001; White, R.E., 2004).

Similarity Searching for Symbolic Data

Methods for MIDI representation

In symbolic music data, the features of music are available in some form, as already previously mentioned, while their extraction reduces in n-dimensional strings of the desired n features. Accordingly, the current literature has long used string processing techniques and indices for similarity searching in symbolic music data. Research is oriented in both monophonic and polyphonic symbolic data.

A number of approximate string matching techniques for musical data, both monophonic and polyphonic have been extensively studied (Clifford, R. & Iliopoulos, C.S. 2004; Cambouropoulos, E., Crochemore, M., Iliopoulos, C.S., Mouchard, L., & Pinzon Y.J., 1999; Crawford, T., Iliopoulos, C.S., & Raman, R., 1998). Despite the fact that approximate string matching has widely been utilised in various fields of computer science, the approximation methods were not devised for the peculiarities of musical data.

Optimal solutions for exact matching proposed by (Crawford, T., Iliopoulos, C.S., & Raman, R., 1998) are the Knuth-Morris-Pratt and variants of the Boyer-Moore algorithms. As far as approximate matching is concerned, (Clifford, R. & Iliopoulos, C.S., 2004) and (Cambouropoulos, E., Crochemore, M., Iliopoulos, C.S., Mouchard, L., & Pinzon Y.J., 1999) propose as optional the δ-, γ- and (δ,γ)-approximation. In δ-approximation "two strings p and t are δ-approximate if and only if they have the same length and each value of p is within δ of its corresponding value in t" (Clifford, R. & Iliopoulos, C.S., 2004). In γ-approximation p and t must have equal length and additionally have a sum of absolute differences, for corresponding values, less or equal to γ. Finally, for p and t to be (δ,γ)- approximate, they must be both δ- and γ- approximate. The best complexity of these algorithms, in general, is $O(nm/w)$, with w being a computer word.

Another approach for approximate matching is δ-, γ- matching using fast Fourier transforms with respective complexities of $O(\delta n \log m)$ and $O(\sigma n \log m)$, σ being the size of alphabet. Additionally, δ-matching can also reduce into two instances of the problem know as *less-than matching* (Amir, A. &

Farach, M., 1995) with complexity $O(\sqrt{mn}\log m)$. In this case, for p and t we require all the values of p be less or equal than the corresponding in t.

String matching techniques have also been proposed for a number of other musicological problems such as approximate matching with gaps, approximate repetitions and evolutionary chains detection (Iliopoulos, C.S., Lemstrom, K., Niyad, M., & Pinzon, Y.J., 2002). In musical patters, reinstatement with certain degree of spontaneity or differentiation is rather common. Thus, searching for notes that do not appear successively is in some cases required. A gap size bounded solution can be solved in $O(nm)$ time (Clifford, R. & Iliopoulos, C.S., 2004). The problem of approximate repetitions is formulated as "Given a music sequence S, an approximate repeating pattern P is an approximate subsequence of S that appears at least twice in S". The exact repetition problem has $O(n \log m)$ complexity and approximate δ-, γ- and (δ,γ)-matching solutions requires $O(n^2)$ time.

In order to address the issue of similarity in polyphony, two main approaches exist. The first method consists of polyphonic reduction to monophony or homophony and accordingly applies monophonic similarity algorithms while in the second special design similarity algorithms are directly implemented on polyphonic data (Liu, N.-H., Wu, Y.-H., & Chen, A.L.P., 2003).

In the special design similarity algorithms for polyphonic data direction, the authors of (Clifford, R. & Iliopoulos, C.S., 2004) opting for exact match of a pattern occurring distributed horizontally in a sequence of notes, proposed the use of a modified Shift-Or algorithm with $O(|\Sigma|+m)+O(N)$ complexity, with $|\Sigma|$ being the number of distinct pitches, m the size of the pattern and N the length of original score. For the case of approximate matching the same work suggests the Modified Wu-Mamber algorithm, with approximate distance, defined using the edit distance for each character.

Research by Szeto et al. (Szeto, W.M. & Wong, M.H., 2003) suggests the extraction of streams based on the musical perception theory that music is perceived in groupings. The proposed methodology separates each musical note according to pitch value and timing into event vectors and following clusters these vectors producing streams. Thus polyphony reduces to the problem of clustering. In (Doraisamy, S. & Ruger, S., 2004) the authors propose an n-gram construction with the use of sliding windows that include events made of pitches with the same or similar onset times. Pickens et al. in (Pickens, J., Bello, J.P., Monti, G., Crawford, T., Dovey, M., Sandler, M. & Byrd, D., 2003) deal with polyphony by ignoring all duration information for every note in the score, and then retaining at each new note onset all the notes that also begin at that onset.

Methods for CMN representation

Most of the up-to-date research in MIR is concerned with music in acoustic and symbolic, MIDI files, representation. However, the symbolic representation, as already mentioned, includes music in the form of notation and especially CMN. The use of CMN is of great importance to music libraries and those musically trained. The number of musical pieces in music notation included solely in the US Library of Congress is believed to be exceeding the six million pieces (Byrd, D., 2001). Thus, the use of mechanical assistance in IR in these collections can be invaluable.

Of the many evolution impeding reasons for the CMN MIR, its complexity and consequently the complexity of the tools required to be built in order to process CMN as well as the unavailability of a standardised format of CMN are the most hampering.

Initial work on the field could only handle simple monophonic music (Maidín, D.Ó., 1998), while in some cases the query had to be in the acoustic format (Bainbridge, D., 1998). Commercial applications have been available, such as Finale (www.finalemusic.com), that can perform search by content in CMN, though searching is limited in a single score at each time, as well as in finding the next match for certain Boolean criteria. Latest development in commercial packages, such as Nightingalesearch (Byrd, D., 2001), overcome the previously mentioned impediments, while offering matching based on pitch and/or duration with approximate matches, under certain tolerance conditions. Though, Nightingalesearch has numerous shortcoming the most important of which is that the supported music files are proprietary of the Nightingale (AMNS, 2000) software.

Similarity Searching for Acoustic Data

In similarity searching in acoustic musical data, feature extraction from the music signal produces the required mapping in which similarity functions as well as speed-up indexing schemes operate.

Up-to-date related work on acoustic data - acoustic query content-based MIR systems is limited. The author in (Yang, C., 2002) proposes a spectral indexing algorithm for CBMIR. Its feature-extraction process attempts to identify distinctive notes or rhythmic patterns. The features are used to construct “characteristic sequences”, which in the next step are indexed in a probabilistic scheme, the Locality-Sensitive Hashing (LSH). The LSH scheme allows both false positive and negative matches, which are compensated in a later step based on the uniformity in time of music tempo changes. Experimental results indicate high retrieval accuracy for different similarity types. In (Won, J.-Y., et al., 2004), the authors propose a CBMIR system that is mainly oriented towards servicing different types of queries. The acceptable query types include audio files, common music notation as well as Query-By-Humming (QBH). The MIDI format is used as an intermediate music object representation. The selection of features is called “representative melody” and is register into an M-tree structure, in which melodies are inserted based on their average length and pitch variation together with melody signatures representing the variation pattern. The used distance is a time-warping function. Preliminary results indicate 65% gain in space requirements when using the collection of features instead of the whole melodies.

As far as the work in (Won, J.-Y., et al., 2004) is concerned, its main disadvantage is the assumption that the users' query must include at least one of the parts that they gather in order to create the “representative melodies”. And as this might work for QBH, it might not for a random piece of a music file included in the index, especially for a small one. In addition, polyphonic music transcription is known to be very hard and poor performing (Yang, C., 2002; Pickens, J., 2004). Regarding the work in (Yang, C., 2002), its feature selection mechanism is oriented towards identifying different types of similarity in music pairs. Additionally, the selected features can lead to false negatives, which have to be addressed in a post-processing step. Finally, (Yang, C., 2002) uses a specialised indexing mechanism.

The approach proposed by (Karydis, I., Nanopoulos, A., Papadopoulos, A., & Manolopoulos, Y., 2004) presents a feature extraction method based on the first few DFT coefficients of the audio file (sequence). The extracted features are grouped by Minimum Bounding Rectangles (MBRs) and indexed by means of a spatial access method. Given a range query and some results, the authors present a false alarm resolution method that utilises a reverse order schema while calculating the Euclidean distance of the query and results, in order to avoid costly calculations. Comparative evaluation to an already existing algorithm shows significantly reduction in execution times. The proposed scheme does not introduce false negatives, according to the used similarity model, and, more importantly, it uses general purpose indexes (R-trees), which allow for a direct implementation in existing RDBMSs.

Similarity Searching Methods in P2P Networks

P2P systems are a rapidly developing area. Searching therein, for music information[10], presents additional requirements in comparison to the customary client-server model. The size of the transferred data to propagate and resolve a query, the CPU burden produced at each peer to resolve a query, as well as the underlying structure of the P2P network and the searching scheme adopted are some the most important facts that need be taken into consideration. In particular, even for compressed acoustic data (e.g., mp3, wma) the traffic produced to solely propagate the query by a simple flooding algorithm is prohibitive. Thus, similarity searching in P2P networks preferably develops in more than one step using various granularity samples. That is, when a coarse-grain (small size) representation of the query returns a match, only then does the querying peer send a more fine-grained (and larger in size) query. Additionally, acoustic data also require increased CPU processing in order to perform similarity functions. Since P2P networks typically consist of computers that are utilised otherwise than the P2P application as well, a CPU resource protection must exist in order to ensure that a queried computer is primarily allocating CPU according to its user will (Yang, C., 2003)

[10] Music information exchanged in P2P networks is customarily in acoustic format.

P2P networks can be classified based on the control over data location and network topology in *unstructured*, *loosely structured* and *highly structured* (Li, X. & Wu, J., 2004). Unstructured P2P networks follow no rule in where data is stored while the network topology is arbitrary (Gnutella). Loosely structured P2P networks have both data location and network architecture non-precisely determined (Freenet). Finally, in highly structured networks data storage and network topology are explicitly defined (Chord). What is more, P2P networks can also be classified according to the number of central directories of document locations in *centralised*, *hybrid* and *decentralised*. Centralised networks maintain a central directory in a single location (Napster), hybrid networks maintain more than directories in super-peers (Kazaa) while for the decentralised (Chord) no central directory is kept.

Karydis et al. (Karydis, I., Nanopoulos, A., Papadopoulos, A., & Manolopoulos, Y. 2005) study several similarity searching algorithms for acoustic data in unstructured P2P networks. The searching schemes that are being imposed include brute-force flooding BFS, quantitative probabilistic >RES (Yang, B. & Garcial-Molina, H., 2002), qualitative ISM (Kalogeraki, V., Gunopulos, D., & Zeinalipour-Yazti, D., 2002) and numerous others (Li, X. & Wu, J., 2004). In the case of the >RES algorithm the query peer Q propagates the query q to a subset k of its neighbour peers all of which returned the most results during the last m queries. Thus, searching for similarity initiates from the most probably larger parts of the networks and is followed by the algorithms discussed in the previous section according to the format of the music file. The ISM approach for each query, a peer propagates the query q to the peers that are more likely to reply the query based on a profile mechanism and a relevance rank. The profile is built and maintained by each peer for each of its neighbouring peers. The information included in this profile consists of the t most recent queries with matches, their matches as well as the number of matches the neighbouring peer reported. The relevance rank function is computed by comparison of the query q to all the queries for which there is a match in each profile.

In structured P2P networks the common scenario of searching scheme is based on Distributed Hash Tables (DHT). In such systems each node is assigned with a region in a virtual address space, while each shared document is associated with a value (id) of this address space. Since in highly structured networks data storage and network topology are explicitly defined, a shared document is stored in the node with address space that the document's id fall within. Thus, locating a document requires only a key lookup of the node responsible for the key. Despite the great acceptance of DHT P2P networks (Chord, Pasty, Can, Koorde, Viceroy, etc), hashing does not support range queries.

CLUSTERING

Clustering in music data has contributed techniques to automatically organise collections of music recordings in order to lessen human documentation efforts. Therefore, clustering algorithms are use to detect groups between music pieces in case where further information (e.g., genre, style, etc.) is not available or not required to be predefined, that is, data-driven categorisation of music styles.

Hierarchical Clustering

Hierarchical clustering is renowned for its usage in multidimensional dataset pattern detection. Analysis based on hierarchical clustering is a statistical method for identification of groups of data (clusters), which indicate relative homogeneity, on the basis of measured characteristics. The analysis begins with one piece of data put in a separate cluster and develops by iteratively combining clusters into broader ones, aiming at reduction of their number and finishing, should a desired quantity of clusters is reached.

There are two types of hierarchical clustering algorithms in order to build a tree from an input set S: the *Agglomerative* approach (bottom-up) and the *Divisive* approach (top-down). The former is the most common approach and the process begins with sets of one element that are subsequently merged until S is achieved as the root. In the latter, a recursive partitioning of S occurs until sets of one element are reached.

Recent work on hierarchical clustering for music databases by Lampropoulos et al. (Lambropoulos, A.S. & Tsihrintzis, G.A., 2004) utilises acoustic data, while the extracted features are based on spectral analysis and tempo. In detail, the spectral features extracted are the mean centroid, mean roll-off, mean flux, zero-crossing and short-time energy. The metrics used therein, are Euclidean distance and cosine distance. The clustering algorithm for *n* data points develops in four steps; (1) initially, each data point occupies a cluster of its own; (2) then for a desired number of *k* clusters if the number of available clusters is *k* stop, else find the pair of clusters with the highest similarity value; (3) merge these clusters, decrease cluster number by one and re-compute the distances between the new cluster and all existing clusters; and finally, (4) repeat procedure from step 2, until all items are clustered into a single cluster of size *n*.

A variation of the hierarchical clustering approach is utilised in (Hoos, H.H., Renz, K., & Gorg, M., 2001) in order to reduce search effort by purging some of the data that do not match the query, and more importantly, by identifying promising candidate data. This approach utilises a modified hierarchical clustering resulting in a balanced tree where each node has up to 32 children. In order to additionally speed up the search within this tree, each node stored is assigned with three bit matrices, the entries of which indicate whether the transition probabilities in the probabilistic model for the cluster corresponding to that node exceed a specific value. The use of these matrices supports rapid selection of the most promising sub-cluster at each step at the internal nodes during the search. The introduction of the previously mentioned mechanisms serve in pruning large sections that cannot include an exact match (since the occurrence of transition probabilities are null) as well as guiding searches to promising candidate pieces as fast as possible.

Other Types of Clustering

The remaining categories of clustering, apart from hierarchical, consist of *k*-clustering (partitioning), Self Organizing Maps (SOM) as well as Hybrid solutions. The target of *k*-clustering is the identification of the best set of k clusters centroids assigning each instance to its nearest centroid, a process that additionally determines the structure of the partition. A SOM is group of several connected nodes mapped into a *k*-dimensional space following some specific geometrical topology (grids, rings, lines, etc). The nodes are initially placed at random, while subsequently iterative adjustment occurs based on the distribution of input along the *k*-dimensional space.

Following, some prominent research works that fall within hybrid category previously mentioned are presented. In work by Pienimäki et al. (Pienimäki, A. & Lemström, K., 2004) polyphonic music is segmented into phrases using initially a monophonic reduction that retains only the highest pitch notes and subsequently an existing melodic phrase algorithm. The hierarchical structure proposed therein is an amalgamation of *paradigmatic* (Cambouropoulos, E. & Widmer, G., 2000) and *collection* (Eerola, T., J¨arvinen, T., Louhivuori, J., & Toiviainen, P., 2001) clustering. In paradigmatic clustering each single document inserted is analysed in order to identify inner structure while collection clustering attempts to cluster a given collection of documents. Initial clustering occurs at the pragmatic level, where variants of a common phrase are clustered together based on a similarity matrix, in which distances are measured by harmonic and melodic edit distances. Each document is described using adjacency lists, while each such list is associated with a document and stores results of paradigmatic and surface level analyses of the corresponding document. At the final step, clustering of the whole collection occurs using the adjacency lists.

Another interesting approach in music clustering is proposed by Cooper et al. (Cooper, M. & Foote, J., 2003). The approach suggested therein is based on methods developed for segmenting still images. Initial time-domain analysis transforms the musical data into features, while similarity is based on the cosine distance. The pairwise similarity of all permutations of features for each music file is computed leading to a “partial time-indexed similarity matrix” (Cooper, M. & Foote, J., 2003) for the detection of audio segment boundaries. The following step includes clustering of the calculated segments by means of similarity analysis, which consists of identification of time-separated repeated segments as well as cases of over-segmentation errors. Based on the segmentation boundaries, the full similarity matrix can be estimated. Then, segment clustering occurs based on singular value decomposition. The proposed scheme, instead of computing the full sample-indexed similarity matrix orientates towards segment-level clustering achieving CPU load gain.

Finally, Cilibrasi et al. (Cilibrasi, R., de Wolf, R., & Vitanyi, P., 2003) propose a clustering scheme based on musical feature compression. The features utilised in (Cilibrasi, R., de Wolf, R., & Vitanyi, P., 2003) are note-on, note-off, average volume and modal note extracted from MIDI files. The average volume result stems from the average value of note velocity in each track, while modal note refers to the most often occurring pitch in each track. The similarity measure used (Li, M., Badger, J.H., Chen, X., Kwong, S., Kearney, P., & Zhang, H., 2001; Li, M. & Vitanyi, P.M.B., 2001/2002; Li, M., Chen, X., Li, X., Ma, B. & Vitanyi, P., 2003) is based on Kolmogorov's complexity. In order to cluster the music data, the proposed method consists of computing a phylogeny tree based on the previously mentioned distance between any two music files. The sub-trees of the phylogeny tree, which is made using a modified quartet method, constitute the clusters that are created based on closeness of objects stored therein.

CLASSIFICATION

Similarly to clustering, classification aims at the purpose of grouping similar documents together. The main difference between clustering and classification is that clustering is a fully automated process requiring no preparation steps or maintenance, classification, on the other hand, generally requires manual, before execution, specification of categories and updating these categories as new documents are added to the collection.

Many different features can be used for music classification, e.g., reference features (title and composer), content-based acoustic features (tonality, pitch, and beat), symbolic features extracted from the scores, and text-based features extracted from the song lyrics. In this section we focus on content-based features and music genre classification. The latter concerns the classification of music from different sources w.r.t. genres, in general, and styles in particular.

Classification with Content-Based Features

In content-based classification, physical features such as pitch, duration, loudness and time/spectral domain features as well as perceptual features such as timbre, its salient components and music properties humans perceive in sound are provided to the classification process. The output of a common classification engine may include retrieved music data similar to one or more of the supplied features, based on previous training of the engine on feature classes or by general similarity. Following are two cases of recent research on the area.

The proposed system by Wold et al. (Wold, E., Blum, T., Keislar, D., & Wheaton, J., 1996) utilises as features loudness, pitch, brightness, bandwidth and harmonicity. Initially, n features are extracted from the music file producing an n-dimensional vector. The training of the system can either be done by directly defining constrains to the values of the feature vector, i.e. a specific value for pitch, or by supplying feature vectors and assigning them to a specific class manually. For each manually defined set of class the proposed methodology calculates the mean vector and covariance matrix. Should a new audio file need be classified, its feature vector is calculated, while using the Euclidean distance it is compared to the class's threshold in order to ensure the degree of similarity. In case of mutually exclusive classes, the newly inserted file is inserted to the class with which its distance is the smallest. In order to define the quality measure of the class, the magnitude of the covariance matrix can be used.

An alternative approach by Li (Li, S.Z., 2000) is based on a classification method called the nearest Feature Line (NFL). The NFL, utilises information provided by multiple prototypes per class explored, in contrast to the nearest neighbour (NN) classification in which the prototype is compared to each query individually. As far as the features used to represent the musical data, Li considers perceptual, cepstral as well as their combinations as features. The NFL's key steps are: interpolation or extrapolation of each pair of prototypes belonging to the same class by a linear model and subsequently, generalisation of the prototypes by the feature line passing through the two points. The feature line's role is to provide information about variants of the two sounds. Thus, the prototype's set capacity is expanded. The classification is achieved using the minimum distance between the query's feature point and feature lines.

Musical Genre Classification

Music can be divided into genres in many different ways, while a genre may contain myriad different styles. These classifications are often arbitrary and controversial, and furthermore closely related styles often overlap. Herein we present issues related to genres with respect to classification. In order to categorically describe music, one can use musical genres. Their ability in structuring vast amounts of music available in digital form is rather popular on the Web as well as on non-online collections, thus become important for MIR.

As previously mentioned, the process of genre categorisation in music can be divided into two steps: *feature extraction* and *multi-class classification*. During feature extraction, the system develops a representation of the music data to be classified that will base on in order to perform the subsequent classification. The extracted features need be musically coherent, *compact* in terms of size and *effective* in order to facilitate the classification.

Research reported on music genre classification is rather limited. Li et al. (Li, T., Ogihara, M, & Li, Q., 2003) proposed the use of DWCHs as features (based on wavelet histograms) to represent music and classified them using the *one-versus-the-other* method. Tzanetakis and Cook (Tzanetakis, G., Essl, G., & Cook, P., 2001) proposed a comprehensive set of features for direct modelling of music signals and used them for musical genre classification using *k*-Nearest Neighbours and Gaussian Mixture models.

In work by Deshpande et al. (Deshpande, H., Singh, R., & Nam, U., 2001) Gaussian Mixtures, Support Vector Machines and Nearest Neighbours are used to classify, based on timbral features, music into rock, piano, and jazz. Finally, Soltau et al. (Soltau, H., Schultz, T., & Westphal, M., 1998) proposed a music classification system using a set of abstract features utilising temporal structures, as well as their classification based on artificial neural networks.

Classification Algorithms

A brief overview of basic classification algorithms and related work includes *k*-Nearest Neighbour Algorithm, Naive Bayesian Algorithm, Concept Vector-based Algorithm, Hierarchical algorithms, and Combination algorithms.

k-Nearest Neighbour (k-NN) Algorithm. *k*-NN classification is an instance-based learning algorithm based on a distance function such as the Euclidean distance or cosine. As the algorithm initiates, the *k* nearest neighbours of the training data are computed. Accordingly, for each sample of the test data its similarity to the k neighbours is computed and thus classified to the most similar class.

Naive Bayesian (NB) Algorithm. This algorithm is renowned to document classification due to its commendable performance. The key idea in this case is the joint probabilities of words and categories leading to the probability of a category for a document. The algorithm computes the posterior probability for each class using Bayes rule and becomes assigned to the class with higher probability. The naive appellation is due to the assumption that the conditional probability of a word, for a given category, is assumed independent from the conditional probabilities of other words in the same category.

Concept Vector-based (CB) Algorithm. The CB algorithm's central idea is that within a class, for each set of documents, the concept vector is computed by summation of all vectors in the class and then normalisation of this sum by its 2-norm. For *n* classes in the training data, *n* concept vectors are produced one for each class. Should a new sample need be classified, initially it is normalised by 2-norm and then its cosine similarity with all *n* concept vectors is computed. Accordingly, the new sample is classified to the class that the most similar concept vector belongs to.

Hierarchical algorithms. In hierarchical algorithms a top-down hierarchical structure is utilised. Thus, the classification problem reduces to hierarchical splits in a tree-like structure. The algorithm initiates from the root in order to distinguish classes. At each subsequently step only one child (children) of the parent selected at the previous step are to be considered. Hierarchical algorithms require pre-processing of the hierarchical structure and use additional data structures for hierarchical information storage.

Combination algorithms. Additionally, a number of possible combinations may be produced from the previously mentioned algorithms. The key idea in this case is to amalgamate approaches in order to selectively retain advantages and avoid specific disadvantages usually at the cost of a parameter that matters little to targeted purposes. Some examples include Concept Vector based algorithm & *k*-NN algorithm, Clustering & CB & *k*-NN Cluster algorithm and Clustering & CB & *k*-NN algorithm.

PATTERN DISCOVERY

The discovery of a repeated structure in music data is a pivotal step in music analysis. Such structures play a crucial role in the understanding of the construction of a musical piece in terms of musical motifs and themes. A theme (especially in classical music) is a melody that the composer uses as a starting point for development, which may be repeated in the form of variations. Repeating patterns have been considered as *characteristic signatures* of music objects, which have the notion of a quantitative measure for music similarity (Crawford, T., Iliopoulos, C.S., & Raman, R., 1998).

Algorithms for Repeating Patterns Discovery

A motif is a minimum pattern that is meaningfully independent and complete within a piece of music. The variation extent and the repetition frequency of a theme can differ depending on the composer and the type of music. Recent research has focused on searching motifs using methods that find repeating patterns in symbolic representations of music data (where the pitch information is selected as the main music feature). Given a music object *S*, a repeating pattern *P* is a subsequence of consecutive elements of *S* that appears at least twice in *S* (Hsu, J.L., Liu, C.C., & Chen, A.L.P., 2001).

The mining of repeating patterns is described in (Hsu, J.L., et al., 2001), where two algorithms are proposed for the discovery of non-trivial repeating patterns and feature melody string. The first algorithm uses a correlative matrix for the extraction of repeating patterns (Hsu, J., et al., 1998), while the second is based on a repeating string-join operation. Experimental results in (Hsu, J.L., et al., 2001) indicate the superiority of the latter algorithm towards the correlative matrix approach. Koh and Yu (Koh, J.L. & Yu, W.D.C., 2001) presented a means of mining the maximum repeating patterns from the melody of a music object using a bit index sequence as well as an extension for extraction of frequent note sequences from a set of music objects. Rollandet al. (Rolland, P.Y. & Ganascia, J.-G., 2002) described an algorithm for mining of sequential patterns in music data, which considers several peculiarities of music objects.

Nevertheless, the number of repeating patterns may be very large, a fact that burdens their examination by human analysts. Existing research has identified that among the collection of repeating patterns, the longest ones are those that can be characterised as feature melody strings and are typically those that can yield to themes. Karydis et al. (Karydis, I., Nanopoulos, A., & Manolopoulos, Y., 2005) proposed an efficient algorithm for finding the longest repeating patterns, which discovers them by using a fast ascending searching procedure, as far as the length of the patterns is concerned, so as to quickly reach the required patterns. Thus, this algorithm avoids the examination of a large number of intermediate patterns and only considers those patterns that are necessary in order to reaching the maximum-length patterns.

Algorithms for Music Theme Discovery

Having argued at the previous section the efficiency and semantic quality of the repeating patterns as far as content-based music data retrieval is concerned, their use in indexing music sequences for the purposes of MIR (Hsu, J.L., et al., 2001), comes as no surprise. Most importantly though, they provide a reference point for the discovery of music *themes* (Liu, C.C., Hsu, J.L., & Chen, A.L.P., 1999; Smith, L. & Medina, R., 2001). Themes, being the musical phrases most likely to be remembered by listeners, make a theme index focus the search on the parts of the database most apt to match a query. Although, a theme should be identified by the previous section, the difficulty that arises is how to distinguish the theme of all repeating patterns discovered. To address this issue a number of theme discovering algorithms that have been proposed, are subsequently presented.

Thus, as far as the use of repeating patterns in theme discovery is concerned, Smith and Medina (Smith, L. & Medina, R., 2001) proposed a pattern matching technique leading to theme discovery that is based on a collection of previously found longest repeating patterns. Meek and Birmingham in (Meek, C. & Birmingham, W.P., 2001) identify numerous features that need be extracted from each music object for the discovery of themes. Among them, they considered as most important the position of the theme (favouring the themes appearing earlier in the music object). As described, such features can be used for the discovery of themes from the repeating patterns found. In addition, an interesting web-based system for theme discovery is presented in (Kornstadt, A., 1998). Patterns may not only be in one voice (the case of polyphonic music), as a pattern may be distributed across several simultaneously sounding voices. (Iliopoulos, C.S. & Kurokawa, M., 2002) and (Iliopoulos, C.S., Niyad, M., Lenstrom, K., & Pinzon, Y.J., 2002) present a number of different algorithms for the discovery of such patterns, including distributed pattern matching with at most k-differences (motif evolution).

SUMMARY & PERSPECTIVES

We have presented the most significant trends in recent research in the field of music mining. Similarity searching has attracted a lot of attention, because it is related to CBMIR, the most prominent application of music mining. Due to peculiarities of music data, we paid special attention to issues regarding the perception of music. Next, we examined how well-known functionalities, like clustering, classification, and detection of repeating patterns have been applied in music mining. As described, music mining presents unique challenges; thus, the developed methods are quite dissimilar to existing ones from other mining fields.

The prospects of music mining, both in terms of research and applications, seem to be encouraging. Since it is relatively new a research field, it contains several open research issues. To name some important ones: methods for detecting meaningful music units, scalable algorithms (which will also consider disk resident music data), and tools for visualisation *and* audition (which is not required in other mining fields) of the extracted patterns. Music mining can and should expand to new application areas as well. To name some few: (i) Tracing of plagiarism and copyright protection, by using clever similarity searching that will disclose hidden reproduction of original music. (ii) E-commerce of music. Attempts like iTune or iMusic may change the paradigm that music is merchandised. We can envisage environments in which users can interactively search for individual music pieces and create their own compilations. For this task, a user can be assisted by music mining, which will help in finding the desired pieces and others pieces as well, which may be previously unknown to the user and different in terms of genre and style. (iii) The relation with industrial standards. MPEG-7 is an example of an emerging standard, which tries to define a description of audio-based musical content. This can have an impact on the hardware industry as well, since manufacturers of recording devices (video cameras, DVD recorders) may want to include the functionality of automatic indexing of recorded music content (Leman, M. 2002). For all the above reasons, we believe that music mining will grow significantly in the forthcoming years.

REFERENCES

AKoff (2001). AKoff Sound Labs, AKoff Music Composer. URL: http://www.akoff.com/

Amir, A. & Farach, M. (1995). Efficient 2-dimensional approximate matching of half-rectangular figures. *Information and Computation*, vol.118, 1-11.

AMNS (2000). Nightingale. http://www.ngale.com

Arakisoftware (2003). Arakisoftware AmazingMIDI. URL: http://www.pluto.dti.ne.jp/~araki/amazingmidi/

Bainbridge, D. (1998). MELDEX: A Web-based Melodic Index Service. In: Melodic Similarity: Concepts, Procedures, and Applications, *Computing in Musicology*, vol. 11, chapter 12, pages 223-230.

Bello, J. P. (2003). Towards the Automated Analysis of Simple Polyphonic Music: A Knowledge-Based Approach. Ph.D. Thesis, Univ. of London.

Bent, I. (1980). Analysis. *Grove's Dictionary of Music*, Macmillan, London.

Bigand, E., Madurell, F., Tillmann, B., & Pineau, M. (1999). Effect of global structure and temporal organization on chord processing. *Journal of Experimental Psychology: Human Perception and Performance*, 25, 184-197.

Byrd D. & Crawford T. (2002). Problems of Music Information Retrieval in the Real World. *Information Processing and Management*, 38(2), 249-272.

Byrd, D. (2001). Music-notation searching and digital libraries. *Proceedings ACM/ IEEE - CS joint conference on Digital libraries*, pp. 239-246.

Cambouropoulos, E. & Widmer, G. (2000). Automatic motivic analysis via melodic clustering. *Journal of New Music Research 29*(4), 303–317.

Cambouropoulos, E., Crochemore, M., Iliopoulos, C.S., Mouchard, L., & Pinzon Y.J. (1999). Algorithms for computing approximate repetitions in musical sequences. *Proceedings Australasian Workshop on Combinatorial Algorithms*, vol. 3, pp.114-128.

Chen, J.C.C. & Chen, A.L.P. (1998). Query by Rhythm An Approach for Song retrieval in Music Databases, *Proceedings Workshop Research Issues in Data Engineering*, pp.139-146.

Cilibrasi, R., de Wolf, R., & Vitanyi, P. (2003). Algorithmic clustering of music. *The Computing*

Research Repository.
Clifford, R. & Iliopoulos, C.S. (2004). Approximate String Matching for Music Analysis, *Soft Computing*.
Cooper, M. & Foote, J. (2003). Summarizing Popular Music via Structural Similarity Analysis. *Proceedings IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.
Crawford, T., Iliopoulos, C.S., & Raman, R. (1998). String matching techniques for musical similarity and melodic recognition. *Computing in Musicology*, 11, 73-100.
Davy, M. & Godsill, S. J. (2003). Bayesian harmonic models for musical signal analysis. *Proceedings Bayesian Statistics VII,* J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith (Eds.), Oxford University Press.
Dervos, D., Linardis, P., & Manolopoulos, Y. (1997). S-index: a Hybrid Structure for Text Retrieval. *Proceedings ADBIS,* pp.204-209.
Deshpande, H., Singh, R., & Nam, U. (2001). Classification of music signals in the visual domain. *Proceedings of the COST-G6 Conference on Digital Audio Effects*.
Doraisamy, S. & Ruger, S. (2004). A Polyphonic Music Retrieval System Using N-Gram. *Proceedings ISMIR*.
Eerola, T., J¨arvinen, T., Louhivuori, J., & Toiviainen, P. (2001). Statistical features and perceived similarity of folk melodies, *Music Perception* 18, 275–296.
Hoos, H.H., Hamel, K.A., Renz, K., & Kilian, J. (1998). The GUIDO Music Notation Format - A Novel Approach for Adequately Representing Score-level Music. *Proceedings ICMC,* pp.451-454, San Francisco.
Hoos, H.H., Renz, K., & Gorg, M. (2001). GUIDO/MIR - an experimental musical information retrieval system based on GUIDO music notation. *Proceedings International Symposium on Music Information Retrieval*.
Hsu, J., Liu, C., & Chen, A.L.P. (1998). Efficient Repeating Pattern Finding in Music Databases. *Proceedings ACM CIKM*.
Hsu, J.L., Liu, C.C., & Chen, A.L.P. (2001). Discovering Non-Trivial Repeating Patterns in Music Data. *IEEE Transactions on Multimedia*, 3(3), 311-325.
Hubbard, T.L. (1996). Synesthesia-like mappings of lightness, pitch, and melodic interval. *American Journal of Psychology*, 109, 219-238.
Iliopoulos, C.S. & Kurokawa, M. (2002). Exact & Approximate distributed Matching for Musical Melodic Recognition. *Proceedings Convention on Artificial Intelligence and the Simulation of Behaviour*, pp.49-56.
Iliopoulos, C.S., Lemstrom, K., Niyad, M., & Pinzon, Y.J. (2002). Evolution of musical motifs in polyphonic passages. *Proceedings Symposium on AI and Creativity in Arts and Science,* pp. 67-76.
Iliopoulos, C.S., Niyad, M., Lenstrom, K., & Pinzon, Y.J. (2002). Evolution of Musical Motifs in Polyphonic Passages. *Proceedings Convention on Artificial Intelligence and the Simulation of Behaviour*, pp.67-75.
Innovative (2004). Innovative Music Systems IntelliScore. URL: http://www.intelliscore.net/
Kalogeraki, V., Gunopulos, D., & Zeinalipour-Yazti, D. (2002). “A Local Search Mechanism for Peer-to-Peer Networks”, *Procceedings of CIKM*, pp.300-307.
Karydis, I., Nanopoulos, A., & Manolopoulos, Y. (2005). Mining Maximum-Length Repeating Patterns in Music Databases. *Multimedia Tools & Applications*, to appear.
Karydis, I., Nanopoulos, A., Papadopoulos, A., & Manolopoulos, Y. (2004). Audio Indexing for Efficient Music Information Retrieval. *Proceedings International Multimedia Modeling Conference* (accepted for publication).
Karydis, I., Nanopoulos, A., Papadopoulos, A., & Manolopoulos, Y. (2005). Evaluation of Similarity Searching Methods for Music Data in Peer-to-Peer Networks. *International Journal of Business Intelligence and Data Mining.*
Lin, K.-I., Jagadish, H.V., & Faloutsos, C. (1994). The TV-Tree: An Index Structure for High-Dimensional Data. *The VLDB Journal*, 3(4), 517-542.
Klapuri, A. (2004). Signal processing methods for the automatic transcription of music. Ph.D Thesis. Tampere University of Technology.
Koh, J.L. & Yu, W.D.C. (2001). Efficient Feature Mining in Music Objects. *Proceedings of*

Database and Expert Systems Applications, pp.221-231.
Kornstadt, A. (1998). Themefinder: A Web-based Melodic Search Tool. *Computing in Musicology*, 11, 231-236.
Kostek, B. & Wieczorkowska, A. (1997). Parametric Representation of Musical Sounds. *Archive of Acoustics*, 3-26.
Lambropoulos, A.S. & Tsihrintzis, G.A. (2004). Agglomerative Hierarchical Clustering for Musical Database Visualization and Browsing. *Proceedings Hellenic Conference on Artificial Intelligence*, pp.177-186.
Lamont, A. & Dibben, N. (2001). Motivic structure and the perception of similarity. *Music Perception*, 18, 245-274.
Lee, W. & Chen, A.L.P. (2000). Efficient Multi-Feature Index Structures for Music Information Retrieval. *Proceedings SPIE*, pp. 177-188.
Leman, M. (2002). Musical Audio Mining. In: J. Meij (Ed.). *Dealing with the Data Flood: Mining data, text and multimedia*. Rotterdam: STT Netherlands Study Centre for Technology Trends.
Li, M. & Vitanyi, P.M.B. (2001/2002). Algorithmic Complexity. *International Encyclopedia of the Social & Behavioral Sciences*, 376–382.
Li, M., Badger, J.H., Chen, X., Kwong, S., Kearney, P., & Zhang, H. (2001). An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2), 149–154.
Li, M., Chen, X., Li, X., Ma, B. & Vitanyi, P. (2003). The similarity metric. *Proceedings. ACM-SIAM Symposium on Discrete Algorithms*.
Li, S.Z. (2000). Content-based Classification and Retrieval of Audio Using the Nearest Feature Line Method. *IEEE Transactions on Speech and Audio Processing*.
Li, T., Ogihara, M, & Li, Q. (2003). A comparative study on content-based music genre classification. *Proceedings ACM SIGIR Conference on Research and development in Information Retrieval*, pp.282-289.
Li, X. & Wu, J. (2004). Searching Techniques in Peer-to-Peer networks. *Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, and Peer-to-Peer Networks*, CRC Press.
Liu, C.C., Hsu, J.L., & Chen, A.L.P. (1999). An Appriximate string Matching Algorithm for Content-Based Music Data Retrieval. *Proceedings IEEE Multimedia Computing and Systems*, pp.451-456.
Liu, C.C., Hsu, J.L., & Chen, A.L.P. (1999). Efficient Theme and Non-trivial Repeating Pattern Discovering in Music Databases. *Proceedings IEEE International Conference on Data Engineering*, pp.14-21.
Liu, N.-H., Wu, Y.-H., & Chen, A.L.P. (2003). Efficient K-NN search in polyphonic music databases using a lower bounding mechanism", *Proceedings of ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp.163-170.
Lo, Y.L. & Chen, S.J. (2003). Multi-Featured Indexing for Music data. *Proceedings IEEE TICDCSW*.
Maidín, D.Ó & Cahill, M. (2001). Score Processing for MIR. *Proceedings International Symposium on Music Information Retrieval (ISMIR)*.
Maidín, D.Ó. (1998). A Geometrical Difference Algorithm. *Computing in Musicology*, 11, 65-72.
McAdams, S., Vieillard, S., Houix, O., & Reynolds, R. (2004). Perception of musical similarity among contemporary thematic materials in two instrumentations. *Music Perception, 22,* 207–237.
Meek, C. & Birmingham, W.P. (2001). Thematic Extractor. *Proceedings International Symposium on Music Information Retrieval*, pp.119-128.
Owen, S. R. (2000). On the Similarity of MIDI Documents. *Harvard University*.
Papaodysseus, C., Roussopoulos, G., Fragoulis, D., Panagopoulos, Th., & Alexiou, C. (2001). A new approach to the automatic recognition of musical recordings. *Jounal of Acoustical Engineering Society*, 49(1/2), 23-35.
Paraskevas, M. & Mourjopoulos, J. (1996). A Statistical Study of the Variability and Features of Audio Signals. *Audio Engineering Society*.
Pickens, J. (2001). A survey of feature selection techniques for music information retrieval. Technical report, Center for Intelligent Information Retrieval, Departament of Computer Science, University of Massachussetts, 2001.

Pickens, J. (2004). Harmonic Modeling for Polyphonic Music Retrieval. PhD Thesis. University of Massachusetts at Amherst.

Pickens, J., Bello, J.P., Monti, G., Crawford, T., Dovey, M., Sandler, M. & Byrd, D. (2003). Polyphonic Score Retrieval Using Polyphonic Audio Queries: A Harmonic Modeling Approach. *Journal of New Music Research,* 32(2), 223-236.

Pienimäki, A. & Lemström, K. (2004). Clustering Symbolic Music Using Paradigmatic and Surface Level Analyses. *Proceedings International Conference on Music Information Retrieval*, pp. 262-265.

Pikrakis, A., Theodoridis, S., & Kamarotos, D. (2002). Recognition of Isolated Musical Patterns using Hidden Markov Models. *Proceedings of International Conference on Music and Artificial Intelligence*, pp.133-143.

Reiss, J., Aucouturier, J.-J., & Sandler, M. (2001). Efficient multidimensional searching routines for music information retrieval. *Proceedings ISMIR*, pp.163-171.

Roads, C. (1996). The Computer Music Tutorial. *MIT Press*.

Rolland, P.-R. & Ganascia, J.-G. (2002). Pattern Detection and Discovery: The Case of Music Data Mining. *Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery*, pp190-198.

Schellenburg, E.G., Krysciak, A. M., & Campbell, R.J. (2001). Perceiving emotion in melody: Interactive effects of pitch and rhythm. *Music Perception*, 18, 155-171.

Seventh (2004). Seventh String Software, Transcribe! URL: http://www.seventhstring.demon.co.uk/

Shepard, R. (1999). Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics. MIT Press, Cambridge, Massachusetts, Chapter 3.

Shifrin, J., Pardo, B., Meek, C., & Birmingham, W. (2002). HMM-based musical query retrieval. *Proceedings ACM/IEEE-CS Conference on Digital libraries*, pp.295-300.

Smith, L. & Medina, R. (2001). Discovering Themes by Exact Pattern Matching. *Proceedings Annual International Symposium on Music Information Retrieval*, pp.31-32.

Soltau, H., Schultz, T., & Westphal, M. (1998). Recognition of music types. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Subrahmanian, V.S. (1998). Multimedia Database systems. *Morgan Kaufmann Publishers*, San Francisco.

Szeto, W.M. & Wong, M.H. (2003). A Stream Segregation Algorithm for Polyphonic Music Databases. *Proceedings IDEAS*, pp.130-138.

Tzanetakis, G., Essl, G., & Cook, P. (2001). Automatic Musical Genre Classification of Audio Signals. *Proceedings International Symposium on Music Information Retrieval*.

Velivelli, A., Zhai, C., & Huang, T.S. (2003). Audio Segment Retrieval Using a Synthesized HMM. *Proceedings ACM SIGIR Workshop on Multimedia Information Retrieval*.

White, R.E. (2004). Recognition and Perception of Whole-Tone, Natural Minor, Harmonic Minor, and Melodic Minor Scales. *Proceedings Annual Convention of the Southwestern Psychological Association*, 2004.

Wieczorkowska, A. & Ras, Z. (2001). Audio Content Description in Sound Databases. *Web Intelligence: Research and Development*, pp.175-183, 2001.

Wieczorkowska, A. (2001). Musical Sound Classification based on Wavelet Analysis. *Fundamenta Informaticae*, 47(1/2), 175-188.

Wold, E., Blum, T., Keislar, D., & Wheaton, J. (1996). Content-based classification, search and retrieval of audio. *IEEE Trans. Multimedia*, 3(3), 27-36.

Won, J.-Y., Lee, J.-H., Ku, K., Park, J., & Kim, Y.-S. (2004). A Content-Based Music Retrieval System Using Representative Melody Index from Music Databases. P*roceedings ADBIS*.

Yang, B. & Garcial-Molina, H. (2002). Improving Search in Peer-to-Peer Networks. *Procceedings Inernational Conference of Distributed Computer Systems*, pp.5-15.

Yang, C. (2002). Efficient Acoustic Index for Music Retrieval with Various Degrees of Similarity. *Proceedings of ACM Multimedia Conference*, pp. 584-591.

Yang, C. (2003). Peer-to-peer architecture for content-based music retrieval on acoustic data. P*roceedings International Conference on World Wide Web*, pp.376-383.