Efficient \(k\)NN search in polyphonic music databases using a lower bounding mechanism

Received: date / Published online: date
© Springer-Verlag 2005

Abstract Querying polyphonic music from a large data collection is an interesting topic. Recently, researchers have attempted to provide efficient methods for content-based retrieval in polyphonic music databases where queries are polyphonic. However, most of them do not work well for similarity search, which is important to many applications. In this paper, we propose three polyphonic representations with the associated similarity measures and a novel method to retrieve \(k\) music works that contain segments most similar to the query. In general, most of the index-based methods for similarity search generate all the possible answers to the query and then perform exact matching on the index for each possible answer. Based on the edit distance, our method generates only a few possible answers by performing the deletion and/or replacement operations on the query. Each possible answer is then used to perform exact matching on a list-based index, which allows the insertion operations to be performed. For each possible answer, its edit distance to the query is regarded as a lower bound of the edit distances between the matched results and the query. Based on the \(k\)NN results that match a possible answer, the possible answers that cannot provide better results are skipped. By using this mechanism, we design a method for efficient \(k\)NN search in polyphonic music databases. The experimental results show that our method outperforms the previous methods in efficiency. We also evaluate the effectiveness of our method by showing the search results to the musician and nonmusician user groups. The experimental results provide useful guidelines on the design of a polyphonic music database.

Keywords Polyphonic music information retrieval · Indexing methods · Search process · Lower bounded edit distance

1 Introduction

The amount of music data in digital formats has been increasing with advances in computer and network technologies. An efficient method for music-content-based retrieval is therefore required. Since it is hard for users to specify precise queries, a music retrieval system should be equipped with the ability to perform similarity search. In addition to music retrieval, the techniques of similarity search can also be used in various applications such as music variation search, copy detection, and music recommendation (music matching the user’s interest is recommended to the user). Similarity search is difficult because the similarity between music pieces depends on human perception, which can be very subjective.

To provide content-based retrieval on music data in symbolic forms, a variety of approaches [22] have been proposed to address the issues of data representation and retrieval efficiency. Several representations of music data [22, 27] have been introduced, such as pitch, rhythm, interval, chord, and contour. To deal with the efficiency issue, different techniques [5] have been presented in the literature, including the methods of string matching, dynamic programming, \(n\)-gram indexing, list-based indexing, and tree-based indexing.

Ghias et al. [10] propose a way to represent music objects, where a music object is represented as a string composed of three kinds of symbols, “U”, “D”, and “S.” Each of the symbols denotes that a note is higher than, lower than, or equal to its previous note. The problem of music retrieval is then transformed into one of approximate string matching. McNab et al. [18] use dynamic programming techniques to find the matches of the given melodic phrases. Issues of melody transcription and the relationship between the matching criteria and the retrieval effectiveness are discussed.

In [3, 6], a system supporting the content-based navigation of music data is presented. A sliding window is applied to cut a music contour into subcontours. All the subcontours are organized as an index for the navigation. Tseng [28] proposes a content-based retrieval model for music collections.
The system uses the pitch profile encoding method for music representation and the n-gram indexing method for similarity search. In [29], a framework is proposed in which the music objects are also organized via the n-gram indexing method for efficient retrieval. The n-gram indexing method is also used in [8, 32, 33]. Furthermore, Downie and Nelson [8] evaluate the effectiveness of the n-gram indexing method via statistical analysis.

Most previous studies focused on the monophonic music. Recently, researchers have attempted to provide efficient methods for content-based retrieval in polyphonic music databases where queries are also polyphonic. This kind of application requires methods that can perform similarity search. For example, a user may want to find a classic music work by an audio segment. By the audio recognition methods for music data [2, 20, 23, 25], the symbolic representation of this audio segment may contain errors. A fault-tolerant retrieval method is therefore required.

Two types of approaches are commonly used for content-based retrieval on polyphonic data. The first type of approach is to transform polyphony into monophony and then apply the techniques to content-based retrieval on monophonic data. For example, Themefinder [12] represents each polyphonic datum by monophonic themes. Given a query, similar themes are retrieved using string matching algorithms. Meldex [16] transforms each polyphonic datum into a monophonic one using the highest-pitch approach proposed by Uitdenbogerd and Zobel [29]. The common drawback in this type of approach lies in its assumption that polyphonic data can be correctly represented as monophonic data, which is not always true [22].

The second type of approach is to design specialized algorithms for content-based retrieval on polyphonic data. SEMENTEX [14] extends the well-known shift-or algorithm [31] for string matching to find all the segments in the polyphonic data that match the monophonic query. This system also enables searching with transposition invariant. To enable polyphonic querying, PROMS [4] adopts the inverted file to record the information of each note in the polyphonic data. The recorded information includes pitch, bar, and note onset time. It only returns for the music segments with exactly the same pitches and onset times of the notes as those in the query. This system is good for exact matching but ineffective for approximate matching because all the onset times of the notes in the query have to match those in the music segment. On the other hand, Dovey [7] proposes a matrix-based algorithm to process the polyphonic query. In addition to exact matching, this algorithm further takes approximate matching into consideration. However, due to the limitation of its matrix representation, its performance suffers from both the long length and the large number of music data.

Unlike the previous methods using the technique of string matching for music retrieval, some approaches employ the technique of music modeling. Due to the complex nature of polyphonic music, most of the approaches using this technique operate in the monophonic domain [9]. A notable exception is the one proposed by Pickens and Crawford [24], where HMM (hidden Markov model) is used to describe the probability distributions of the chords in polyphonic data and polyphonic queries. The probability distributions are represented as probability matrices. In this way, the distance between a query and a datum can be computed according to the difference between their probability matrices. Such an approach is appropriate for music with variations. However, the probability distribution of a string can be very different from the probability distribution of its substring. Therefore, the HMM-based approach can only be used for cases where the queries and data have almost the same lengths.

Given two strings, a variety of scoring functions can be used to estimate the similarity between them. Given a query string and a scoring function, two approaches are often used to select proper answers. One is to select the data whose scores are above a predefined threshold. This approach is commonly named range querying or epsilon search. The other is to select only a fixed number of data whose scores are the highest (named the k nearest-neighbors search, abbreviated as kNN search). This approach is more suitable for music retrieval since only a predetermined number of music works are returned to the user.

The set of notes that begin at the same time is defined as an event in [7]. Therefore, a piece of polyphonic music can be regarded as a string of events. Owing to the variety of events, conventional methods such as the suffix tree [17, 21] and the n-gram indexing [8] often generate a complex and costly index, and thus the kNN search can be inefficient. Methods for music reduction [22] have been proposed to reduce the complexity of the index by grouping similar events together. However, a change in the similarity measure for events will result in the reconstruction of the index, which is still time consuming.

In this paper, we propose three polyphonic representations with the associated similarity measures and a novel method to retrieve k music works that contain segments most similar to the query. An overview of our method is illustrated in Fig. 1. Based on the edit distance, our method generates only a few possible answers, called the expanded queries (EQs), by performing deletion and/or replacement operations on the query. Each EQ is then used to perform exact matching on a list-based index, which allows the insertion operations to be performed. For each EQ, its edit distance to the query is regarded as a lower bound of the edit distances between the matched results and the query. Based on

---

**Fig. 1** An overview of our method