Variations on Local Alignment for Specific Query Types

Alexandra L. Uitdenbogerd
RMIT University
Melbourne, Victoria
Australia
sandrau@rmit.edu.au

Abstract

For this submission to MIREX, we again provided a simple baseline for comparison with other submissions. For short incipit queries we used a form of dynamic programming on melody strings that enforces matching from the start of the strings — a technique we call *Start-Match Alignment*. There is some evidence that this technique is better than local alignment for short queries and incipits.

For the query by humming track we used both local alignment and Start-Match Alignment. It seems that for this collection, query set and relevance set, Start-Match Alignment works better than local alignment on average. Our algorithms were the fastest of those submitted for the symbolic search tasks, and for the polyphonic symbolic task had very good effectiveness.

Keywords: melody matching, music retrieval

1. Introduction

Our entry in the Symbolic Melodic Similarity (SMS) and the Query by Singing/Humming (QBSH) tracks of the 2006 round of MIREX used one of the techniques shown in currently unpublished work to be more effective than local alignment under some circumstances. We call this particular matching technique "Start-Match Alignment", as it finds the best match between two strings, where the match is enforced to commence at the start of the strings.

The start-match technique is applied atop the basic melody extraction and melody standardisation techniques of our three-stage melody matching model [2]. It is our belief that this combination of techniques leads to robust matching. With the application of current efficient programming methods, the approach can be used to produce a practical system for purely symbolic matching. Our techniques had yet to be tested on hummed queries, so it was interesting to see that the techniques don't appear to work as well as those that are optimised for such queries.

2. Techniques

The family of music matching techniques our team have developed are based around the 3-stage melody matching model: melody extraction, melody standardisation, and melody matching. For monophonic queries this simplifies to just the second and third stage of the model.

In this submission we use the melody extraction technique *allmono* originally developed and tested in earlier work [3, 4, 5]. Similarly, we use the *directed modulo-12* approach to standardisation, which is a convenient simplification of pitch interval strings into a smaller representation that can be mapped into alphabetic characters (a concept used by Hawley in earlier work [1]). Since the representation is purely based on pitch, variations in tempo between a sung query and the target piece of music will not affect the ranking.

The melody matching technique used for this submission is a little different to our earlier work. Based on work that is currently in submission, the approach enforces matching of strings from the start of the strings until a best matching length is found. This technique, which we have named *Start-Match Alignment*, initialises and fills the array in the manner of global alignment, but, in the manner of local alignment, returns the highest score within the matrix. The equation used to calculate each cell's value is the same as for global alignment.

$$a[i,j] = max \begin{cases} a[i-1,j]+d & i \geq 1 \\ a[i,j-1]+d & j \geq 1 \\ a[i-1,j-1]+e & p(i) = t(j) \text{ and } i,j \geq 1 \\ a[i-1,j-1]+m & p(i) \neq t(j) \text{ and } i,j \geq 1 \\ 0 & i,j = 0 \end{cases} \quad (1)$$

where d is the cost of an insert or delete, e is the value of an exact match, m is the cost of a mismatch, i and j are non-negative integers, $p(i)$ represents the ith symbol in the "pattern" or query, and $t(j)$ represents the jth symbol in the "text", or potential answer string. The weights we used were 1 for a match, -1 for a mismatch, and -2 for an insert or delete (indel).

Each melody string in the collection is compared to the query string, with ranking based on the computed alignment score. The higher the alignment score, the more similar the

two strings are assumed to be.

For QBSH we chose to use the above algorithm in addition to standard local alignment of the standardised melody strings. Local alignment locates the best substring match between two strings, regardless of where the match occurs and its length.

Due to a slip-up in the script preparation, our submission for the SMS tasks normalised the score by dividing by the log of the length of the track (in number of symbols plus one) against which the query was being compared. Results on our collection suggest that this is slightly less effective than not normalising. See the Discussion section for possible reasons.

3. MIREX Tasks and Results

SMS task 1 consisted of incipit queries matched against the RISM collection of incipits. Half of the queries were transcriptions of a hummed or whistled source. Both collection and queries were monophonic. In this task our technique was ranked in the middle of the set of algorithms. It was one of three techniques that used an "indexing" phase. The query time was the fastest, being 31 seconds, with the next shortest being 59 seconds. At 64 seconds, the sum of the indexing and querying times was 4 seconds shorter than the fastest query-phase-only submission.

SMS task 2 involved search for melodies in a polyphonic collection. Of the five algorithms, Start-Match was ranked second across various measures of effectiveness. The query time was faster than all other submissions, as was the indexing time.

The QBSH tasks used a very large collection of sung queries against a monophonic collection consisting of 48 ground truth quantised melodies in addition to the Essen collection of folk songs. For this task our best algorithm performed poorly compared to other submissions (eighth out of thirteen for task 1 and second last for task 2). However, it is clear that the Start-Match algorithm was more effective than local alignment. Once again our submission was very fast compared to other entrants.

4. Discussion

This particular submission was not optimised for speed. Retaining the collection strings in memory, or using a compressed form for matching would be much faster, as would the use of a heap for retaining the top 10 results. Despite the above, the submission was the fastest at answering queries in both tasks of the SMS track.

As mentioned earlier, the SMS submission inadvertently applied normalisation to the alignment scores. When tested on our own collection this leads to slightly worse effectiveness. On spending considerable time examining the retrieved answers to the queries, it became clear that a reason normalisation may be unhelpful is that the length of the *track* was used, and not the length of the entire piece. This can lead to short bits of accompaniment getting much higher scores than warranted. Despite this liability, our submission was in the middle range for effectiveness in SMS task 1, and the second most effective algorithm for task 2.

For the QBSH tasks, there was an obvious difference in effectiveness results between those algorithms working directly with the WAV files, and those using the provided MIDI. There may have been errors introduced into the MIDI files at the time of manual transcription (as suggested by the organiser), which made the task more difficult. While our technique was the fastest for these tasks, the use of MIDI files instead of WAV meant that less processing was required than some of the other submissions, making some comparisons meaningless.

The lack of success in task 2 of QBSH for Start-Match may in part be attributed to the types of errors found in sung queries. The alignment approach we used causes two symbols to be incorrect when a single note error occurs in a query, such as a wrong note substituted for a correct note. This can cause the penalty to be too great when matching some query-melody pairs. There are more robust techniques for this alignment problem that we intend to apply in the future.

An issue that seems apparent is that algorithms work best on collections and queries most similar to those that researchers have used for their development. Our algorithms were developed on monophonic symbolic queries against a polyphonic collection, and they worked best on these. Typke et al.'s development has largely been on incipit collections, and this led to excellent results for that domain. Jang et al.'s familiarity with the use of sung queries appeared to pay off in the QBSH track. Rather than being an issue of fair-play, it highlights what is known about training and test collections. It also makes clear that the assumptions about the nature of the collection and queries results in different choices for optimal algorithms.

5. Conclusion

MIREX 2006 evaluation has shown that a simple alignment technique applied to pitch interval strings is still competitive for some symbolic query and collection types, but doesn't handle sung queries quite as well as other approaches that were submitted. Future work for our team should include experimentation with sung query sets.

6. Acknowledgments

We thank Iman S. H. Suyoto for providing his MidiPerl code from MIREX 2005.

References

[1] M. Hawley. The personal orchestra, or audio data compression by 10,000:1. *Computing Systems*, 3(2):289–329, 1990.

[2] A. L. Uitdenbogerd and J. Zobel. An architecture for effective music information retrieval. *J. American Society of Information Science and Technology (JASIST)*, 55(12):1053–1057.

[3] A. L. Uitdenbogerd and J. Zobel. Manipulation of music for melody matching. In B. Smith and W. Effelsberg, editors, *Proc. ACM International Multimedia Conference*, pages 235–240, Bristol, UK, September 1998. ACM, ACM Press.

[4] A. L. Uitdenbogerd and J. Zobel. Melodic matching techniques for large music databases. In D. Bulterman, K. Jeffay, and H. J. Zhang, editors, *Proc. ACM International Multimedia Conference*, pages 57–66, Orlando Florida, USA, November 1999. ACM, ACM Press.

[5] A. L. Uitdenbogerd and J. Zobel. Music ranking techniques evaluated. In M. Oudshoorn, editor, *Proc. Australasian Computer Science Conference*, Melbourne, Australia, January 2002.