Musical Alignment Using Globally Optimal Short-Time Dynamic Time Warping

Hagen Kaprykowsky[1], Xavier Rodet[2]

[1] *Deutsche Forschungszentrum für Künstliche Intelligenz GmbH, 67663 Kaiserslautern, Germany, Email: kaprykowsky@iupr.dfki.de*

[2] *IRCAM, 75004 Paris, France, Email: xavier.rodet@ircam.fr*

Abstract

Dynamic Time Warping (DTW) aligns two sequences by time warping them optimally. Global optimization is done using whole sequences. This can be very demanding in terms of calculation costs and memory requirements which means the sequence length that is possible to align is limited. In this paper a novel algorithm Short-Time Dynamic Time Warping (STDTW) is presented, which requires much less memory because optimization is done iteratively on smaller portions of the sequences. The particularly remarkable characteristic of the algorithm is that it finds the same globally optimal solution, under some weak hypothesis as the classical DTW algorithm. As an example, STDTW is applied to Musical Alignment which links events in a musical score and points on a audio performance time axis. It also provides an interesting insight into the structure of the sequences to be aligned.

Introduction

Score to audio alignment is the association of events in a musical score (in our case notes) with points in the time axis of an audio signal. An alignment implies a segmentation of the performance according to the events in the score. The audio performance is a digital recording (a signal) of the score played by musicians, which is referred to as the performance. Here the score is represented by a Standard MIDI File (SMF). This is not a trivial task since the exact tempo is not known and not constant, the notes are never played exactly as written in the score, and finding notes in a complex polyphonic performance is extremely difficult. The performance audio signal can be coded as a sequence of frequency analysis data along time, and the score is a sequence of notes and chords.

Dynamic Time Warping

A classical approach for musical alignment is the use of Dynamic Time Warping (DTW [1]) which finds the best alignment (or match), between two (supposedly time) sequences. The alignment is considered as a non-linear "warping" of one (or the other) sequence along "time". Suppose we have two sequences X and Y of elements x_m and y_n and of length M and N respectively: $X = x_1, x_2, \ldots, x_m, \ldots, x_M = \{x_m, m = 1, \ldots, M\}$, $Y = y_1, y_2, \ldots, y_n, \ldots, y_N = \{y_n, n = 1, \ldots, N\}$. The alignment of an element x_{mk} with an element y_{nk} is defined by a couple $a_k = (m_k, n_k)$, $1 \leq m_k \leq M$ and $1 \leq n_k \leq N$. An alignment of one sequence with the other is defined by a sequence $A = a_1, a_2, \ldots, a_k, \ldots, a_K$ such that the sequences $\{m_k, k = 1, \ldots, K\}$ and $\{n_k, k = 1, \ldots, K\}$ are non decreasing ($m_{k-1} \leq m_k, n_{k-1} \leq n_k$). If one considers a plane with indices m on the abscissa and n on the ordinate, A also defines a "path" in this plane. To define the best alignment, a (local) distance between an element x_{mk} and an element y_{nk}, $d(x_{mk}, y_{nk})$, is first chosen. To simplify notation, let us denote this distance as $d(m_k, n_k)$. Then, the global distance along an alignment or path A is the sum of the weighted local distances along A, i.e. between the sequence elements which are aligned:

$$D(X, Y, A) = \sum_{k=1}^{K} (w_k \times d(m_k, n_k)) \quad (1)$$

where the w_k are weights to be defined in the following. The optimal alignment A_O is the one which minimizes the global distance $D(X, Y, A)$.

Dynamic Programming

There are exponentially many warping paths that satisfy the above conditions. The DTW algorithm first calculates the augmented distance array adm of size MxN where $adm(m, n)$ is the cost of the best path from $(1, 1)$ up to (m, n). The value $adm(m, n)$ is computed recursively by using the local distance $d(m, n)$, of the weights w and of the values $adm(i, j)$ in a neighborhood (i, j) of (m, n) with $i < m$ and $j < n$. Dynamic programming is a fast way to find the optimal path, which minimizes the global distance, i.e. the total warping cost $adm(M, N)$. It relies on the Viterbi algorithm with complexity $O(n^2)$ applied on the adm array. This is done in a backtrack final step that very simply computes optimal predecessors from (M, N) to $(1, 1)$

Path Constraints

Possible paths A are limited by several constraints. Endpoint Constraints force the warping path to start and finish in the opposite diagonal corner of the rectangle (m, n). Monotonicity Constraints force the points a_k to be monotonically placed in time. Local Continuity Constraints force the possible steps, in the warping path, to adjacent cells (including diagonally adjacent cells). Different types of local constraints exist and the weights along the local path branches can be tuned in one direction or another if needed (see [1] for details). For score to audio alignment, type V was found to be the best. Type V constraints the mean slope of path A to be between 3 and 1/3. The standard values for the local path weights $[w_v \ w_h \ w_d]$ are [1 1 2] for type V. In our experiments, to reduce the computation time and the resources needed, at every iteration m we keep only

the best paths by pruning the paths which have an augmented distance $adm(m,n)$ greater than a given threshold θ_p. This threshold is dynamically set using the minimum of the previous adm row. Global path constraints and path pruning finally define a corridor which limits the possible paths. From now on we will only consider paths which are optimal from $(1,1)$ to a certain (m,n), and we will call them paths without mentioning this restriction.

Short-Time Dynamic Time Warping

In general it is not possible to find the globally optimal alignment without calculating all accumulated distances and saving all corresponding predecessors. However, it is clear that the entire score and the entire performance are not necessary to perform a correct alignment. Most of the time, a shorter portion is enough, provided it contains a sufficient number of notes and cannot be mistaken for another (unambiguity). There are short-time versions of DTW that works iteratively on shorter portions. But the limits of these portions in the score and in the performance are obviously not known since the alignment is not known. Also, while DTW finds the globally optimal alignment, these short-time versions do not guarantee that the short-time solution is the globally optimal one. To fulfill these requirements, we have designed a new algorithm, called STDTW (Short-Time Dynamic Time Warping), which relies on a short-time alignment and, under some weak hypothesis, provides the same globally optimal solution as the classical global DTW algorithm. STDTW iteratively works on a short-time portion of the sequences and finds, however, at each iteration a portion of the optimal path. This means that, after each iteration, the computation can be stopped, the path stored and the algorithm restarted from the last point of the path, i.e. on a portion of the two sequences as if they were new sequences.

Approach

With respect to the initial constraints, it is obvious that all paths have to have a common point, at least at $(1,1)$. In figure 1 one can see all backtracks in the corridor from a given m (audio frame index). There appears to be a portion common to all paths from point $(1,1)$ up to the so called fusion point where all paths fuse. There is only one path which goes from $(1,1)$ to the fusion point. This path is therefore a part of the optimal path from (M,N) to $(1,1)$, so it is globally optimal. Now suppose one path cannot cross another one; then it is sufficient to only calculate the two backtracks from rmin and rmax to determine the fusion point and obtain a part of the global optimal path from $(1,1)$ to the fusion point. For type V it does not seem obvious that paths cannot cross each other. A proof can be found in [2].

Method

In order to determine a part of the global optimal alignment it is sufficient to determine the fusion point of the two backtracks. The path from $(1,1)$ to the fusion point

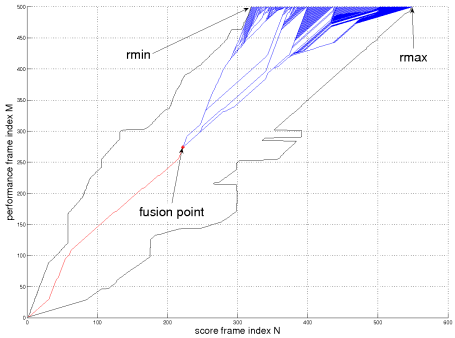

Figure 1: Backtracks from a fixed audio frame index

is a part of the optimal path, because the backtracks cannot cross each other. The algorithm proceeds with increasing values of one of the indices, say $m = 1, 2, 3, \ldots$. In order to diminish the cost of two backtracks and of the fusion point determination, these are done only every $mStep$, e.g. $mStep = 100$, by the modulo instruction below.

For $m = 1$ **to** M

- Calculation of $adm(m,n)$ and storage of the optimal predecessor of (m,n).
- $If(modulo(m, mStep)) = 0$
  - Calculation of the backtracks from rmin and rmax and determination of the fusion point.
  - At this point it is only necessary to keep the predecessors inside the last backtracks and the part of the optimal path which has been determined up to the fusion point. All the previous data can be cleared.

endFor

Conclusion

The new Short-Time Dynamic Time Warping algorithm theoretically permits the alignment of a performance of arbitrary length, and to obtain the same globally optimal result as the global DTW algorithm. It is possible to reinitialise the algorithm at the last fusion point that was determined. This provides the possibility of stopping the algorithm, as well as to restart the calculation (if some system problem occurred for instance) without having to recalculate the whole alignment. It also provides an interesting insight into the structure of the sequences to be aligned.

References

[1] Rabiner, L. R. and Juang, Biing-Hwang : Fundamentals of Speech Recognition, Prentice Hall, Englewood Cliffs, 1993.

[2] Kaprykowsky H. and Rodet X.: Globally Optimal Short-Time Dynamic Time Warping Applications to Score to Audio Alignment, in IEEE ICASSP, 2006, Toulouse