Partial Elastic Matching of Time Series

Longin Jan Latecki Vasileios Megalooikonomou Qiang Wang Rolf Lakaemper
Computer and Information Science Dept., Temple University
Philadelphia, PA 19094, USA
{ latecki, vasilis, lakamper, qwang }@temple.edu

C. A. Ratanamahatana E. Keogh
Computer Science and Engineering Dept., University of California
Riverside, CA 92521
{ ratana, eamonn }@cs.ucr.edu

Abstract

We consider the problem of elastic matching of time series. We propose an algorithm that determines a subsequence of a target time series that best matches a query series. In the proposed algorithm we map the problem of the best matching subsequence to the problem of a cheapest path in a DAG (directed acyclic graph). The proposed approach allows us to also compute the optimal scale and translation of time series values, which is a nontrivial problem in the case of subsequence matching.

1 Introduction and Related Work

Time series are a ubiquitous and increasingly prevalent type of data, therefore, there has been much research effort devoted to time series data mining in recent years. Many data mining algorithms have similarity measurement at their core. Examples include motif discovery [3], anomaly detection [7], rule discovery [6], classification [11] and clustering [1]. In this paper we deal with computation of time series distances based on elastic matching.

As many researchers have mentioned in their work [6, 11, 10], the Euclidean distance is not always the optimal distance measure for similarity searches. For example, in some time series, different parts have different levels of significance in their meaning. Also, the Euclidean distance does not allow shifting in time axis, which is not unusual in real life applications.

To solve the problem of time scaling in time series, Dynamic Time Warping (DTW) [13, 2] aligns the time axis prior to the calculation of the distance. The DTW distance between time series is the sum of distances of their corresponding elements. Dynamic programming is used to find corresponding elements so that this distance is minimal. The DTW distance has been shown to be superior to the Euclidean in many cases [1, 8, 4, 15] (see [12] for a detailed discussion of DTW). As illustrated in Section 2, DTW requires the matched time series to be well aligned, and it is particularly sensitive to outliers, since it is not able to skip any elements of the target series. DTW always matches the query time series to the whole target time series.

The Longest Common Subsequence (LCSS) measure has been used in time series [5, 14] to deal with the alignment and outliers problems. Given a query and a target series, LCSS determines their longest common subsequence, i.e., LCSS finds subsequences of the query and target (of the same length) that best correspond to each other. The distance is based on the ratio between the length of longest common subsequence and the length of the whole sequence. The subsequence does not need to consist of consecutive points, the order of points is not rearranged, and some points can remain unmatched. When LCSS is applied to time series of numeric values, one needs to set a threshold that determines when values of corresponding points are treated as equal [14]. The performance of LCSS heavily depends on correct setting of this threshold, which may be a particularly difficult problem for some applications.

The proposed MVM (Minimal Variance Matching) algorithm computes the distance value between two time series directly based on the distances of corresponding elements, just as DTW does, and it allows the query sequence to match to only a subsequence of the target sequence, just as LCSS does.

The main difference between LCSS and MVM is that LCSS optimizes over the length of the longest common subsequence (which requires the distance threshold), while MVM directly optimizes the sum of distances of corresponding elements (without any distance threshold). The

main difference between DTW and MVM is that MVM can skip some elements of the target series when computing the correspondence while DTW requires that each point of the query sequence is matched to each element of the target sequence. LCSS allows skipping elements of both the query and the target sequence. Therefore, MVM should be used when one is interested in finding the best matching part of the target sequence for a given query sequence, since it guarantees that the whole query sequence will be matched. This is, for example, the case, when the query is a model sequence, one wants to find in a given data set. However, when the query sequence contains outliers and skipping them is allowed, then LCSS should be used.

2 Motivation

For many datasets we can easily and accurately extract the beginning and ending of patterns of interest. However in some domains it is non-trivial to define the exact beginning and ending of a pattern within a longer sequence. This is a problem because if the endpoints are incorrectly specified they can swamp the distance calculation in otherwise similar objects. For concreteness we will consider an example of just such a domain and show that Minimal Variance Matching (MVM), proposed in this paper, can be expected to outperform Dynamic Time Warping (DTW) and Euclidean distance. There is increasing interest in indexing sports data, both from sports fans who may wish to find particular types of shots or moves, and from coaches who are interested in analyzing their athletes performance over time. Let us consider the high jump. We can automatically collect the athletes' center of mass information from video and convert it to time series. In Fig. 1, we see 3 time series automatically extracted from 2 athletes.

Both sequence **A** and **B** are from one individual, a tall male, and **C** is from a (relatively) short female with a radically different style. The difference in their technique is obvious even to a non-expert, however **A** and **C** where automatically segmented in such a way that the bounce from the mat is visible, whereas in **B** this bounce was truncated. In Fig. 1(*middle*) we can see that DTW is forced to map this bounce section to the end of sequence **B**, even though that sequence clearly does not have a truly corresponding section. In contrast, MVM is free to ignore the sections that do not have a natural correspondence. It is this difference that enables MVM to produce the more natural clustering shown in Fig. 1(*bottom*). While this is a somewhat contrived example on a specialized domain, similar remarks apply to many commercially important domains including medical data mining and oil exploration.

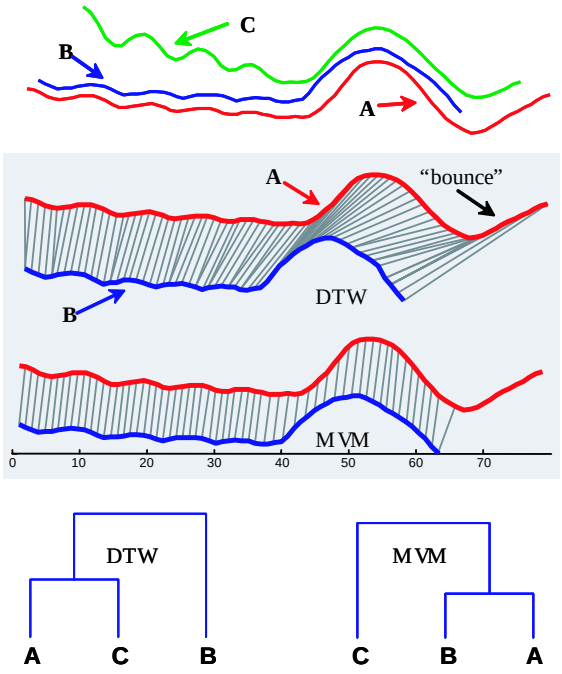

Figure 1. (*top*) Three examples of athletes trajectories as they attempt a high jump. The sequence shows the height of their center of mass (with possible parallax effects). Reading left to right we can see their bounding run followed by the takeoff and landing. (*middle*) The alignment achieved by DTW and MVM on two of the sequences. (*bottom*) The clustering achieved by DTW and MVM.

3 Minimal Variance Matching

We now present an algorithm for elastic matching of two time series of different lengths m and n, which we will call **Minimal Variance Matching (MVM)**. More specifically, for two finite sequences of real numbers $a = (a_1, ..., a_m)$ and $b = (b_1, ..., b_n)$ with $m < n$, the goal is to find a subsequence b' of b of length m such that a best matches b'. Thus, we want to find the best possible correspondence of sequence a to a subsequence b' of b. Formally we define a **correspondence** as a monotonic injection

$$f : \{1, ..., m\} \rightarrow \{1, ..., n\},$$

(i.e., a function f such that $f(i) < f(i+1)$) such that a_i is mapped to $b_{f(i)}$ for all $i \in \{1, ..., m\}$. The set of indices $\{f(1), ..., f(m)\}$ defines the subsequence b' of b. Recall that in the case of DTW, the correspondence is a relation on the set of indices $\{1, ..., m\} \times \{1, ..., n\}$, i.e., a one-to-many and many-to-one mapping.

Once the correspondence is known, it is easy to compute the distance between the two sequences. We do not have any restrictions on distance functions, i.e., any distance function is possible. To allow for comparison to the existing time series matching techniques, we use the Euclidean distance in this paper:

$$d(a,b,f) = \sqrt{\sum_{i=1}^{m}(b_{f(i)} - a_i)^2}. \quad (1)$$

Our goal is to find a correspondence f so that $d(a,b,f)$ is minimal. More precisely, an optimal correspondence $\hat{f}$ of numbers in series a to numbers in series b is defined as the one that yields the global minimum of $d(a,b,f)$ over all possible correspondences f:

$$\hat{f} = \text{argmin}\{d(a,b,f) : f \text{ is a correspondence}\}. \quad (2)$$

Finally, the optimal distance is obtained as

$$d(a,b) = d(a,b,\hat{f}) = \sqrt{\sum_{i=1}^{m}(b_{\hat{f}(i)} - a_i)^2}. \quad (3)$$

In other words $d(a,b)$ is the global minimum over all possible correspondences.

We can also state the correspondence problem in a statistical framework. Let us assume that there is a subsequence b' of b that is a noisy version of a such that $a \sim b' - \mathcal{N}(0,v)$, where $\mathcal{N}(0,v)$ denotes a zero-mean Gaussian noise variable with variance v, i.e., $b' = (b_{f(i)})_i$ for $i \in \{1,...,m\}$.

Since the mean of the differences $(a_i - b_{f(i)})_i$ is zero, i.e., $a - b' \sim \mathcal{N}(0,v)$, the variance σ^2 of difference sequence $(a_i - b_{f(i)})_i$ is given by

$$\sigma^2(a,b,f) = \frac{1}{m}\sum_{i=1}^{m}(b_{f(i)} - a_i)^2. \quad (4)$$

Clearly, $\sigma^2(a,b,f) = v$ (the variance of the Gaussian noise). Observe that in this case the variance corresponds to the Euclidean distance (1). Thus, the variance of the difference sequence is minimal when mapping f establishes a correct correspondence of elements of both sequences.

Now we describe the method used to minimize (4). We first form the difference matrix $r = (r_{ij}) = (b_j - a_i)$. It is a matrix with m rows and n columns with $m < n$. For example, the difference matrix for two time series t_1=(1, 2, 8, 6, 8) and t_2=(1, 2, 9, 3, 3, 5, 9) is shown in Fig. 2. Observe that t_1 and t_2 are similar if we ignore the two elements in t_2 with value 3.

Clearly, (r_{ij}) can be viewed as a surface over a rectangle of size m by n, where the height at point (i,j) is the value r_{ij}. We obtain the correspondence with minimal variance

$$r = \begin{bmatrix} \boxed{0} & 1 & 8 & 2 & 2 & 4 & 8 \\ -1 & \boxed{0} & 7 & 1 & 1 & 3 & 7 \\ -7 & -6 & \boxed{1} & -5 & -5 & -3 & 1 \\ -5 & -4 & 3 & -3 & -3 & \boxed{-1} & 3 \\ -7 & -6 & 1 & -5 & -5 & -3 & \boxed{1} \end{bmatrix}$$

Figure 2. In order to compute $\hat{f}$ for t_1=(1, 2, 8, 6, 8) and t_2=(1, 2, 9, 3, 3, 5, 9), we first form the difference matrix with rows corresponding to elements of t_1 and columns to elements of t_2.

by solving the least-value path problem on the difference matrix. To obtain the solution, we treat (r_{ij}) as a directed graph with the following links:

r_{ij} is directly linked to r_{kl} if and only if **(1)** $k - i = 1$ and **(2)** $j < l$. When traversing the obtained directed graph, the meaning of both conditions is as follows. For any two consecutive points r_{ij}, r_{kl} in each path (1) means that we always go to the next row, while (2) means that we can skip some columns, but cannot go backwards.

We want to have a least-value path with respect to the following cost function for each directed link, $linkcost(r_{ij}, r_{kl}) = (r_{kl})^2$, under the restrictions in (1), (2), and the following ones: Each path can start in first row, between columns 1 and $n - m$, i.e., at r_{1j} for $j = 1,...,n-m$ and the path can end at r_{mj} for $j = n-m,...,n$. The conditions (1) and (2) imply that we can obtain a DAG (directed acyclic graph) G whose nodes are the elements of $(r_{ij})_{ij}$ and weights are defined by the function $linkcost$. It is well known that we can solve the least-value path problem using the shortest path algorithm on G. The obtained least-value path defines exactly correspondence $\hat{f}$, which minimizes (4) in accordance with (2).

The shortest path for the example matrix in Fig. 2 is marked with boxes. Following the boxes, the optimal correspondence $\hat{f}$ is given by

$$\hat{f}(1) = 1,\ \hat{f}(2) = 2,\ \hat{f}(3) = 3,\ \hat{f}(4) = 6,\ \hat{f}(5) = 7.$$

Finally, from (3) we obtain the distance $d(t_1,t_2) = \sqrt{3} \approx 1.732$.

The path computed this way gives us correspondence $\hat{f}$ with the smallest variance of the differences of the corresponding pairs. We recall that this is true, since we assumed that the mean of the differences of the corresponding pairs is zero. Observe that without this assumption, it would not be possible to use the shortest path algorithm on a DAG.

4 MVM and Shift / Scale Estimation

The definition of MVM presented in Section 3 allows us to estimate the linear transformation that best maps query sequence a to a subsequence of target sequence b. This gives a serious advantage with respect to existing time series matching methods. The estimation is done while computing the correspondence $\hat{f}$, and it does not increase the computational complexity of the algorithm. To focus our attention, we present here the estimation of the translation (shift) of values of time series b. The estimation of the scaling factor can be computed analogously.

Now for two finite sequences of real numbers $a = (a_1, ..., a_m)$ and $b = (b_1, ..., b_n)$ with $m < n$, the goal is to find a subsequence b' of b of length m (i.e., correspondence $\hat{f}$) and a translation tr such that a best matches $b' + tr$. This means that we want to minimize:

$$d(a, b, f, t) = \sqrt{\sum_{i=1}^{m}((b_{f(i)} + tr - a_i))^2}. \qquad (5)$$

Observe that if a matches to the whole sequence b, a simple normalization of values of both sequences solves the translation problem. However, this is not the case when a matches only to part of b as we described in the introduction.

Let f_k be any correspondence from $a = (a_1, ..., a_k)$ to $b = (b_1, ..., b_n)$ with $k < m$. Then we can estimate the translation for f_k as

$$tr(a, b, f, k) = \sum_{i=1}^{k} b_{f(i)} - a_i. \qquad (6)$$

The main idea of the solution to (5) is the fact that we can update $tr(a, b, f, k)$ incrementally as

$$tr(a, b, f, k+1) = \frac{k}{k+1} tr(a, b, f, k) + \frac{1}{k+1}(b_{f(k+1)} - a_{k+1}). \qquad (7)$$

By integrating this incremental update in the process of computation of the cheapest path on DAG, we obtain an optimal solution to (5).

5 Conclusions and Future Work

The proposed new method for time series matching, called MVM, performs the following tasks simultaneously **(1)** automatically determines whether the query sequence best matches the whole target sequence or only part of the target sequence, **(2)** automatically skips outliers that are present in the target sequence, **(3)** computes the translation or scale of corresponding values that minimizes the statistical variance of dissimilarities of corresponding elements. By mapping the problem of elastic matching of sequences to the problem of finding a cheapest path in a DAG, we provide an efficient algorithm to compute MVM. Experimental results demonstrating that this method outperforms DTW and LCSS are presented in [9].

References

[1] J. Aach and G. Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17:495–508, 2001.

[2] D. Berndt and J. Clifford. Using dynamic time warping to fi nd patterns in time series. In *Proc. AAAI-94 W. on Knowledge Discovery and Databases*, pages 229–248, 1994.

[3] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Washington, 2003.

[4] S. Chu, E. Keogh, D. Hart, and M. Pazzani. Iterative deepening dynamic time warping for time series. In *Proc. SIAM Int. Conf. on Data Mining*, 2002.

[5] G. Das, D. Gunopoulos, and H. Mannila. Finding similar timie series. In *Proc. 1st PKDD Symposium*, pages 88–100, 1997.

[6] F. Höppner. Discovery of temporal patterns. learning rules about the qualitative behavior of time series. In *Proc. European Conf. on Principles and Practice of Knowledge Discovery in Databases*, pages 192–203, Freiburg, 2001.

[7] E. Keogh, S. Lonardi, and C. Ratanamahatana. Towards parameter-free data mining. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Seattle, 2004.

[8] G. Kollios, M. Vlachos, and D. Gunopoulos. Discovering similar multidimensional trajectories. In *Proc. Int. Conf. on Data Engineering*, pages 673–684, San Jose, 2002.

[9] L. J. Latecki, V. Megalooikonomou, Q. Wang, R. Lakaemper, C. A. Ratanamahatana, and E. Keogh. Elastic partial matching of time series. In *Conf. PKDD*, 2005.

[10] V. Megalooikonomou, Q. Wang, G. Li, and C. Faloutsos. A multiresolution symbolic representation of time series. In *Proc. IEEE Int. Conf. on Data Engineering (ICDE05)*, pages 668–679, Tokyo, 2005.

[11] D. Rafi ei. On similarity-based queries for time series data. In *Proc. Int. Conf. on Data Engineering*, pages 410–417, Sydney, 1999.

[12] C. A. Ratanamahatana and E. Keogh. Everything you know about dynamic time warping is wrong. In *W. on Mining Temporal and Sequential Data*, Seattle, 2004.

[13] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[14] M. Vlachos, M. Hadjieleftheriou, D. Gunopoulos, and E. Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proc. of ACM SIGKDD*, pages 216–225, Washington, 2003.

[15] B. Yi, K. Jagadish, and C. Faloutsos. Effi cient retrieval of similar time sequences under time warping. In *Proc. Int. Conf. on Data Engineering*, pages 23–27, 1998.