

Exact indexing of dynamic time warping

Eamonn Keogh, Chotirat Ann Ratanamahatana

University of California–Riverside, Computer Science and Engineering Department, Riverside, USA

Abstract. The problem of indexing time series has attracted much interest. Most algorithms used to index time series utilize the Euclidean distance or some variation thereof. However, it has been forcefully shown that the Euclidean distance is a very brittle distance measure. Dynamic time warping (DTW) is a much more robust distance measure for time series, allowing similar shapes to match even if they are out of phase in the time axis. Because of this flexibility, DTW is widely used in science, medicine, industry and finance. Unfortunately, however, DTW does not obey the triangular inequality and thus has resisted attempts at exact indexing. Instead, many researchers have introduced approximate indexing techniques or abandoned the idea of indexing and concentrated on speeding up sequential searches. In this work, we introduce a novel technique for the exact indexing of DTW. We prove that our method guarantees no false dismissals and we demonstrate its vast superiority over all competing approaches in the largest and most comprehensive set of time series indexing experiments ever undertaken.

Keywords: Dynamic time warping; Indexing; Lower bounding; Time series

1. Introduction

The indexing of very large time series databases has attracted the attention of the database community in recent years. The vast majority of work in this area has focused on indexing under the Euclidean distance metric (Agrawal et al. 1995; Chan et al. 2003; Das et al. 1998; Debregeas and Hebrail 1998; Faloutsos et al. 1994; Keogh et al. 2000, 2001; Korn et al. 1997; Yi and Faloutsos 2000). However, there is an increasing awareness that the Euclidean distance is a very brittle distance measure (Aach and Church 2001; Bar-Joseph et al. 2002; Berndt and Clifford 1994; Chu et al. 2002; Diez and Gonzalez 2000; Kadous 1999; Keogh and Pazzani 2000; Kollios et al. 2002; Schmill et al. 1999; Yi et al. 1998). What is needed is a method that allows an elastic shifting of the time axis, to accommodate sequences that are similar but out of phase, as shown in Fig. 1. Just such a technique, based on dynamic

Received 10 February 2003

Revised 12 June 2003

Accepted 18 December 2003

Published online 13 May 2004

programming, has long been known to the speech-processing community (Itakura 1975; Kruskal and Liberman 1983; Myers et al. 1980; Rabiner and Juang 1993; Rabiner et al. 1978; Sakoe and Chiba 1978; Tappert and Das 1978). Berndt and Clifford introduced the technique, dynamic time warping (DTW), to the database community (Berndt and Clifford 1994). Although they demonstrate the utility of the approach, they acknowledge that its resistance to indexing is a problem and that “... performance on very large databases may be a limitation.” Despite this shortcoming of DTW, it is still widely used in various fields. In bioinformatics, Aach and Church successfully applied DTW to RNA expression data (Aach and Church 2001). In chemical engineering, it has been used for the synchronization and monitoring of batch processes in polymerization (Gollmer and Posten 1995). DTW has been successfully used to align biometric data, such as gait (Gavrila and Davis 1995), signatures (Munich and Perona 1999) and even fingerprints (Kovacs-Vajna 2000). Many researchers, including Caiani et al. (1998), have demonstrated the utility of DTW for ECG pattern matching. Rath and Manmatha have applied DTW to the problem of indexing repositories of handwritten historical documents (Rath and Manmatha 2002) (although handwriting is two-dimensional, it can trivially be rerepresented as a one-dimensional time series). Finally, in robotics, Schmill et al. demonstrated a technique that utilizes DTW to cluster an agent’s sensory outputs (Schmill et al. 1999).

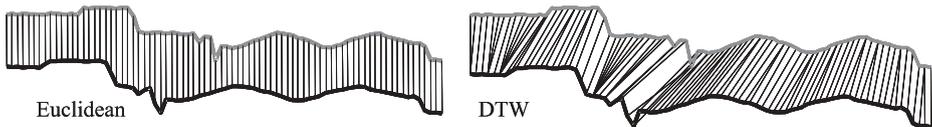


Fig. 1. Note that, while the two time series have an overall similar shape, they are not aligned in the time axis. Euclidean distance, which assumes the i^{th} point in one sequence is aligned with the i^{th} point in the other, will produce a pessimistic dissimilarity measure. The nonlinear dynamic time warped alignment allows a more intuitive distance measure to be calculated

Although applications listed above are quite diverse, one thing they have in common is that they need to find the best match to a query time series, from a (possibly very large) pool of candidates. This can trivially be achieved by sequential scanning, comparing each and every candidate to the query, in an arbitrary order. The problem with sequential scanning is that it is simply too slow for most applications. What we really need is a technique to *index* the data, that is, to find the best match without having to examine every candidate (Roussopoulos et al. 1995; Seidl and Kriegel 1998). Indexing techniques can be *exact*, guaranteeing to return the same result as sequential scanning (Faloutsos et al. 1994; Keogh et al. 2000, 2001), or *approximate*, returning good matches but not necessarily the best matches (Faloutsos and Lin 1995; Park et al. 2001; Yi et al. 1998).

More than two-dozen techniques have been introduced to index time series under the Euclidean distance (Chan et al. 2003; Faloutsos et al. 1994; Keogh et al. 2000; Yi and Faloutsos 2000) (see Keogh et al. (2001) for a more comprehensive listing). In addition, several researchers have shown techniques to approximately index DTW (Park, personal communication) or introduced methods to reduce its demanding CPU time (Berndt and Clifford 1994; Chu et al. 2002). However, only two researchers have claimed to introduce an exact indexing technique for DTW (Kim et al. 2001; Park et

al. 1999). In the case of Park et al. (1999), the claim of no false dismissals was later retracted (Park et al. 1999, 2001). We have carefully implemented the only technique to correctly claim the ability to exactly index DTW and the only other lower bounding approximation of DTW for detailed comparison with our proposed approach.

In contrast with the approaches above, we will prove the no-false-dismissal property of our approach and demonstrate its superiority with the most comprehensive set of time series indexing experiments ever undertaken. In particular, in terms of number and diversity of datasets, size of datasets, range of query lengths and indexing parameters, our experiments are one to two orders of magnitude more than all previous papers combined.

The rest of the paper is organized as follows. In Sect. 2, we will consider the utility of time series similarity search, review the DTW algorithm, and consider related work. In Sect. 3, we will introduce a novel lower bounding technique that tightly approximates the true DTW distance. Section 4 introduces a method that allows the exact indexing using our lower bounding function. In Sect. 5, we conduct an exhaustive empirical comparison of our method with competing techniques. Finally, in Sect. 6, we offer conclusions and suggestions for extensions.

2. Background

A similarity search in time series is useful in its own right as a tool for interactive exploration of very large databases; it is also a subroutine in many data mining applications, including rule discovery (Das et al. 1998), clustering (Debregeas and Hebrail 1998) and classification (Diez and Gonzalez 2000; Kadous 1999). The superiority of DTW over Euclidean distance for these tasks has been demonstrated by several authors (Aach and Church 2001; Bar-Joseph et al. 2002; Caiani et al. 1998; Chu et al. 2002; Keogh and Pazzani 2000; Yi et al. 1998). However, for completeness, we include a simple experiment to illustrate the point.

The most studied time series classification/clustering problem is the cylinder–bell–funnel dataset (Diez and Gonzalez 2000; Kadous 1999); it is a deceptively simple-looking three-class problem. All classes are of length 128. The **cylinder** class consists of a short flat section, followed by a sudden jump to an elevated plateau, and a return to a short flat section. Both the location of the onset of the plateau and the length of the plateau itself are controlled by random variables. The **bell** class replaces the plateau with a ramp and the **funnel** class is simply the mirror image of the **bell** class. Finally, all instances are corrupted by Gaussian noise. More details about the dataset and a data generator can be downloaded from the UCR Time Series Data Mining Archive (<http://www.cs.ucr.edu/~eamonn/TSDMA>). Figure 2 shows typical examples of each class.

The problem has been attacked with sophisticated techniques, including rule-based learners (Diez and Gonzalez 2000; Kadous 1999), boosting, Bayesian techniques, and decision trees. We performed a simple classification experiment on this dataset, using the one-nearest neighbor algorithm. Our dataset consists of ten instances of each class and the classifier was evaluated using the leaving-one-out strategy. Because we had the luxury of unlimited data, we averaged the results over 1,000 runs. The mean error rate for the Euclidean distance metric on the problem was 0.2734, but for DTW, it was only 0.0269, an order of magnitude lower. This off-the-shelf result is competitive with the highly tuned, sophisticated techniques enumerated above. The lower error rate came with a cost, however; classification with DTW took approximately 230 times longer than that with Euclidean distance.

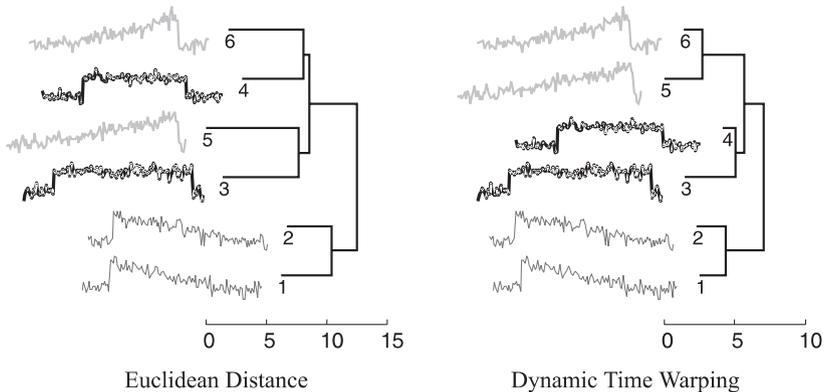


Fig. 2. Typical examples from the cylinder–bell–funnel dataset (cylinder 3 & 4, bell 5 & 6, funnel 1 & 2). When clustered using Euclidean distance, the variability of the time axis often causes the classes to be confused; in contrast, DTW compensates for the variability in the time axis and does a much better job at correctly grouping the classes

Table 1. The basic notation used in this work

C	A time series of length n $C = c_1, c_2, \dots, c_i, \dots, c_n$
$[c_i : c_j]$	A subsequence of C , beginning at c_i and ending at c_j
\bar{C}	A piecewise aggregate approximation of a time series (Keogh et al. 2000; Yi and Faloutsos 2000)
DTW	The dynamic time warp distance measure
LB_Kim	The lower bounding function introduced by Kim et al. (2001)
LB_Yi	The lower bounding function introduced by Yi et al. (1998)
LB_Keogh	The lower bounding function introduced in this work

This result reiterates the utility of DTW and motivates the necessity of indexing it.

2.1. Review of DTW

Suppose we have two time series, Q and C , of length n and m , respectively, where

$$Q = q_1, q_2, \dots, q_i, \dots, q_n \quad (1)$$

$$C = c_1, c_2, \dots, c_j, \dots, c_m. \quad (2)$$

To align two sequences using DTW, we construct an n -by- m matrix where the $(i^{\text{th}}, j^{\text{th}})$ element of the matrix contains the distance $d(q_i, c_j)$ between the two points q_i and c_j (i.e. $d(q_i, c_j) = (q_i - c_j)^2$). Each matrix element (i, j) corresponds to the alignment between the points q_i and c_j . This is illustrated in Fig. 3. A warping path W is a contiguous (in the sense stated below) set of matrix elements that defines a mapping between Q and C . The k^{th} element of W is defined as $w_k = (i, j)_k$. So

we have

$$W = w_1, w_2, \dots, w_k, \dots, w_K \quad \max(m, n) \leq K < m + n - 1. \quad (3)$$

The warping path is typically subject to several constraints.

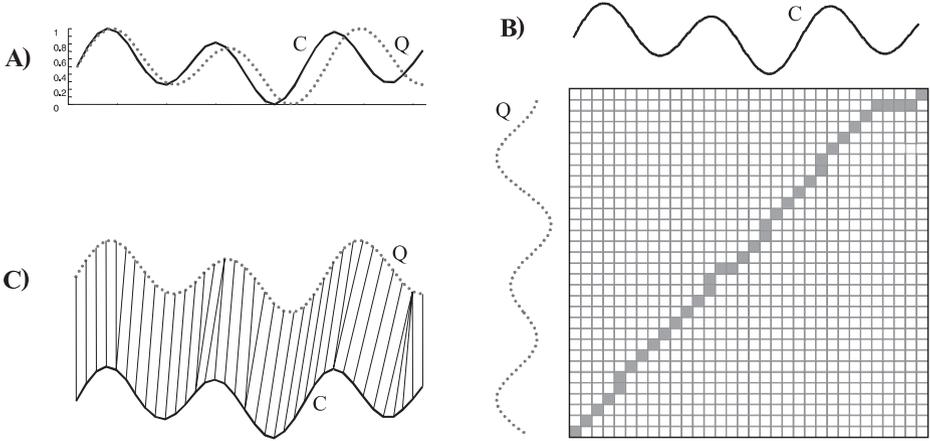


Fig. 3. A) Two sequences Q and C that are similar but out of phase. B) To align the sequences, we construct a warping matrix and search for the optimal warping path, shown with solid squares. C) The resulting alignment

- **Boundary conditions:** $w_1 = (1, 1)$ and $w_K = (m, n)$. This requires the warping path to start and finish in diagonally opposite corner cells of the matrix.
- **Continuity:** Given $w_k = (a, b)$, then $w_{k-1} = (a', b')$, where $a - a' \leq 1$ and $b - b' \leq 1$. This restricts the allowable steps in the warping path to adjacent cells (including diagonally adjacent cells).
- **Monotonicity:** Given $w_k = (a, b)$, then $w_{k-1} = (a', b')$, where $a - a' \geq 0$ and $b - b' \geq 0$. This forces the points in W to be monotonically spaced in time.

There are exponentially many warping paths that satisfy the above conditions. However, we are only interested in the path that minimizes the warping cost:

$$DTW(Q, C) = \min \left\{ \sqrt{\sum_{k=1}^K w_k} \right. \quad (4)$$

This path can be found using dynamic programming to evaluate the following Recurrence, which defines the cumulative distance $\gamma(i, j)$ as the distance $d(i, j)$ found in the current cell and the minimum of the cumulative distances of the adjacent elements:

$$\gamma(i, j) = d(q_i, c_j) + \min \{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \}. \quad (5)$$

The Euclidean distance between two sequences can be seen as a special case of DTW where the k th element of W is constrained such that $w_k = (i, j)_k, i = j = k$. Note that it is only defined in the special case where the two sequences have the same length. The time and space complexity of DTW is $O(nm)$.

This review of DTW is necessarily brief; we refer the interested reader to Kruskal and Liberman (1983) and Rabiner and Juang (1993) for a more detailed treatment.

2.2. Related work

While there has been much work on indexing time series under the Euclidean metric (Chan et al. 2003; Faloutsos et al. 1994; Keogh et al. 2000, 2001; Yi and Faloutsos 2000), there has been much less progress on indexing under DTW.

Yi et al. (1998) introduced a technique for approximate indexing of DTW that utilizes their FastMap technique (Faloutsos and Lin 1995). The idea is to embed the sequences into Euclidean space such that the distances between them are approximately preserved, then classic multidimensional index structures can be utilized (Guttman 1984; Seidl and Kriegel 1988). In addition, they introduced a lower bounding function (described in more detail in Sect. 3.2) that can be used to prune some of the inevitable false hits their method will introduce. The method does produce an observed (maximum) speedup of 7.8 over sequential scanning. However, this does have some limitations. First, it does allow false dismissals. Second, while the time to build the index is linear in M (the size of the database), it is actually $O(Mn^2)$, which quickly becomes intractable for very large databases and/or long sequences.

Kim et al. introduced an exact algorithm for indexing of time series under DTW (Kim et al. 2001). The method extracts four features from the sequences and organizes them in a multidimensional index structure. They introduced a lower bounding function (described in more detail in Sect. 3.2) that is defined on the four features and thus guarantees no false dismissals. Although the work introduced the first technique for exact indexing under DTW, it suffers from several limitations. First, the method only allows the extraction of exactly four features and thus cannot take advantage of multidimensional index structures. In addition, although four features are extracted, only one of them (determined at query time) is actually used in the lower bounding function; thus, the lower bound is very loose and many false alarms are generated, each of which will require evaluation with the quadratic-time DTW algorithm.

In Park et al. (1999), the authors demonstrate a DTW indexing technique that is based on a piecewise linear representation of the data. They prove that this method can guarantee no false dismissals. Unfortunately, the no-false-dismissals claim is incorrect. A candidate sequence in the database can differ from the query sequence by an arbitrarily small epsilon and still not be retrieved (Park, personal communication). A later version of the paper did carry a disclaimer stating, “... *it is possible that a subsequence similar to a query in terms of the original time warping distance may not be included in the answer set in our approach*” (Park et al. 2001). However, this qualification understates the problem. Having tested the approach with 39,200 experiments on 32 different datasets, we found that the approach only returned the true best match to a one-nearest neighbor query 613 times. This result does not significantly differ from random chance. We therefore exclude this approach from further consideration.

Another attempt at indexing DTW utilizes a suffix tree (Park et al. 2000). While the method is interesting, we do not include it in our empirical comparisons because the index size is one to two orders of magnitude larger than the data itself. Such enormous space overhead is simply untenable for very large databases. In any case, the claimed speedup is rather modest.

There has also been some work in which attempts at indexing and/or lower bounding are abandoned, and instead, efforts are concentrated on fast approximation of the DTW distance using a lower resolution approximation of the data. The idea was introduced by Keogh and Pazzani (2000), who use a piecewise linear approximation of the data. The method shows significant speedup with few false dismissals. A similar idea was suggested by Chan et al. (2003). Here, the authors obtain the lower resolution of the data approximation with wavelets and use their approximate distance measure instead of that of Yi et al., i.e. the lower bounding measure, within the FastMap framework. The method improves the speedup of the work of Yi et al. work at the expense of introducing more false dismissals.

Finally, there has been some work on obtaining warping alignments by methods other than DTW (Bar-Joseph et al. 2002; Kwong et al. 1996). For example, Kwong et al. consider a genetic algorithm-based approach (Kwong et al. 1996), and recent work by Bar-Joseph et al. considers a technique based on linear transformations of spline-based approximations (Bar-Joseph et al. 2002). However, both methods are stochastic and require multiple runs (possibly with parameter changes) to achieve an acceptable alignment. In addition, both methods are clearly nonindexable. However, both works do reiterate the superiority of warping over nonwarping for pattern matching.

3. Lower bounding the DTW distance

In this section, we explain the importance of lower bounding and introduce our new lower bounding distance measure.

3.1. The utility of lower bounding measures

Time series similarity search under the Euclidean metric is heavily I/O bound; however, similarity search under DTW is also very demanding in terms of CPU time. One way to address this problem is to use a fast lower bounding function to help prune sequences that could not possibly be the best match. Table 2 gives the pseudo-code for such an algorithm.

Table 2. An algorithm that uses a lower bounding distance measure to speed up the sequential scan search for the query Q

Algorithm Lower_Bounding_Sequential_Scan(Q)
1. $best_so_far = \text{infinity};$
2. for all sequences in database
3. $LB_dist = \text{lower_bound_distance}(C_i, Q);$
4. if $LB_dist < best_so_far$
5. $true_dist = \text{DTW}(C_i, Q);$
6. if $true_dist < best_so_far$
7. $best_so_far = true_dist;$
8. $index_of_best_match = i;$
9. endif
10. endif
11. endfor

There are only two desirable properties of a lower bounding measure:

- It must be fast to compute. Clearly, a measure that takes as long to compute as the original measure is of little use. In our case, we would like the time complexity to be at most linear in the length of the sequences.
- It must be a relatively tight lower bound. A function can achieve a trivial lower bound by always returning 0 as the lower bound estimate. However, in order for the algorithm in Table 2 to be effective, we require a method that more tightly approximates the true DTW distance.

While lower bounding functions for string edit, graph edit, and tree edit distance have been studied extensively (Kruskall and Liberman 1983), there has been far less work on DTW, which is very similar in spirit to its discrete cousins. Below, we will consider the existing DTW lower bounding techniques.

3.2. Existing lower bounding measures

To the best of our knowledge, there are only two existing lower bounding functions available for DTW (not including Park et al. (1999), which incorrectly claims to be lower bounding, or Park et al. (2000), which has a time complexity equal to the full algorithm). While referring the interested reader to the original papers for detailed explanations, below, we give a visual intuition and brief explanation of each.

The lower bounding function introduced by Kim et al. (2001) (hereafter, referred to as LB_Kim), works by extracting a four-tuple feature vector from each sequence. The features are the first and last elements of the sequence, together with the maximum and minimum values. The maximum squared differences of corresponding features are reported as the lower bound. Figure 4 illustrates the idea.

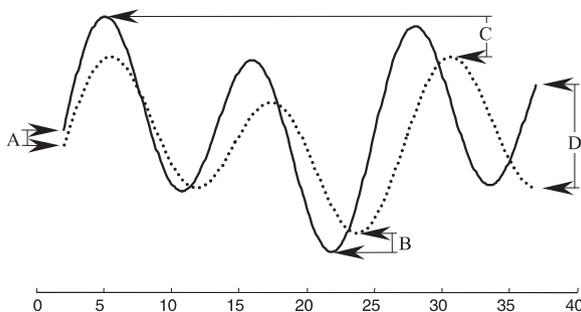


Fig. 4. A visual intuition of the lower bounding measure introduced by Kim et al. The maximum squared difference between the two sequences first (A), last (D), minimum (B) and maximum points (C) is returned as the lower bound

The lower bounding function introduced by Yi et al. (1998) (hereafter, referred to as LB_Yi) takes advantage of the observation that all the points in one sequence that are larger (smaller) than the maximum (minimum) of the other sequence must contribute at least the squared difference of their value and the maximum (minimum) value of the other sequence to the final DTW distance. Figure 5 illustrates the idea.

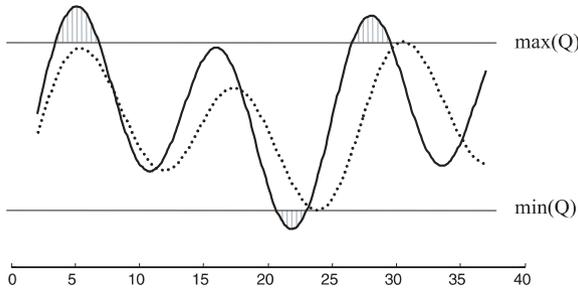


Fig. 5. A visual intuition of the lower bounding measure introduced by Yi et al. The sum of the squared length of gray lines represents the minimum of the corresponding points contribution to the overall DTW distance, and thus can be returned as the lower bounding measure

3.3. Proposed lower bounding measure

Before introducing our lower bounding technique, we must review additional details of the DTW algorithm that we deliberately omitted until now.

3.3.1. Global constraints on time warping

In addition to the constraints on the warping path enumerated in Sect. 2.1, virtually all practitioners using DTW also constrain the warping path in a global sense by limiting how far it may stray from the diagonal (Berndt and Clifford 1994; Chu et al. 2002; Gollmer and Posten 1995; Itakura 1975; Keogh and Pazzani 2000; Myers et al. 1980; Sakoe and Chiba 1978; Tappert and Das 1978). The subset of the matrix that the warping path is allowed to visit is called the warping window. Figure 6 illustrates two of the most frequently used global constraints, the Sakoe-Chiba band (Sakoe and Chiba 1978) and the Itakura parallelogram (Itakura 1975).

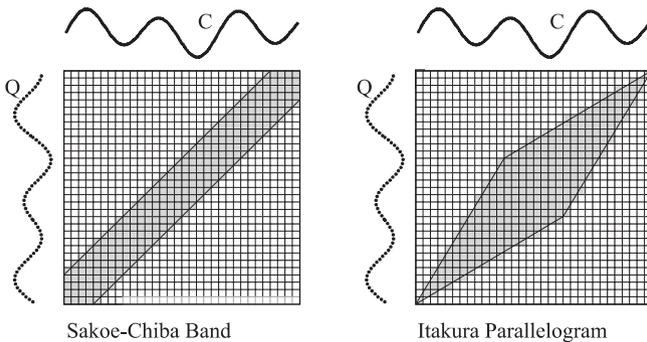


Fig. 6. Global constraints limit the scope of the warping path, restricting them to the gray areas. The two most common constraints in the literature are the Sakoe-Chiba band and the Itakura parallelogram

There are several reasons for using global constraints, one of which is that they slightly speed up the DTW distance calculation. However, the most important rea-

son is to prevent pathological warpings, where a relatively small section of one sequence maps onto a relatively large section of another. The importance of global constraints was documented by the originators of the DTW algorithm, who were exclusively interested in aligning speech patterns (Sakoe and Chiba 1978). However, it has been empirically confirmed in other settings, including finance, medicine, biometrics, chemistry, astronomy, robotics, and industry.

3.3.2. Local constraints on time warping

In addition to the global constraints listed above, there has been active research on local constraints (Itakura 1975; Myers et al. 1980; Rabiner and Juang 1993; Sakoe and Chiba 1978; Tappert and Das 1978) for several decades. The basic idea is to limit the permissible warping paths, by providing *local* restrictions on the set of alternative steps considered. For example, we can visualize Eq. (5) as a diagram of admissible step patterns, as in Fig. 7a. The lines illustrate the permissible steps the warping path may take at each stage. We could replace Eq. (5) with $\gamma(i, j) = d(i, j) + \min \{\gamma(i-1, j-1), \gamma(i-1, j-2), \gamma(i-2, j-1)\}$, which corresponds with the step pattern shown in Fig. 7c. Using this equation, the warping path is forced to move one diagonal step for each step parallel to an axis. The effective intensity of the slope constraint can be measured by $P = n/m$. Figures 7.a to 7.d illustrate the four original constraints suggested by Sakoe and Chiba (1978); in addition, many others have been suggested, including asymmetric ones. The constraint might be designed based on domain knowledge, or from experience learned through trial and error. Rabiner and Juang’s classic paper contains an extensive review (Rabiner and Juang 1993). The important implication of local constraints for our work is the fact that they can be reinterpreted as global constraints. Figure 7.e shows an example.

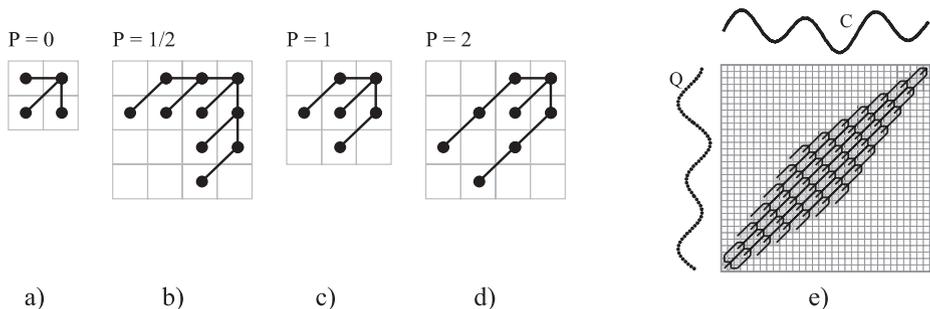


Fig. 7. a to d Four local constraints on dynamic time warping, as suggested by Sakoe and Chiba. **a**) corresponds to the trivial case of no constraint, and is therefore equivalent to Eq. (5), $\gamma(i, j) = d(i, j) + \min \{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}$. In contrast, **c**) corresponds to $\gamma(i, j) = d(i, j) + \min \{\gamma(i-1, j-1), \gamma(i-1, j-2), \gamma(i-2, j-1)\}$. Local constraints can be reinterpreted as global constraints, as an example, **d**) can be reinterpreted as the global constraint shown in **e**), which looks superficially like the Itakura parallelogram constraint

To reinterpret a local constraint as a global constraint, we can do the following. Create a shadow matrix, which is the same size as the DTW matrix. Initialize all the elements of the shadow matrix as unreachable. Call the DTW function, using the relevant constraint; every time the recurrence visits a new cell (i, j) in the original

DTW matrix, the cell (i, j) of the shadow matrix can be labeled as reachable. The convex hull of all the reachable cells forms a band, which can be interpreted as a global constraint. Note that we only have to do this once and we can then store the resulting constraint for future use. While this reinterpretation of local constraints may be obvious, we state it explicitly because it has not appeared in the literature to our knowledge. Finally, we note that global and local constraints can be used together; the interpretation being that, where they conflict, the most restrictive constraint (i.e. the one that forces the path closest to the diagonal line) is used (Kruskall and Liberman 1983).

3.3.3. Proposed lower bounding measure

We can view a global or local constraint as constraining the indices of the warping path $w_k = (i, j)_k$ such that $j - r \leq i \leq j + r$, where r is a term defining the *reach*, or allowed range of warping, for a given point in a sequence. In the case of the Sakoe-Chiba band, r is independent of i ; for the Itakura parallelogram, r is a function of i .

We will use the term r to define two new sequences, U and L:

$$U_i = \max(q_{i-r} : q_{i+r}) \tag{6}$$

$$L_i = \min(q_{i-r} : q_{i+r}). \tag{7}$$

U and L stand for *Upper* and *Lower*, respectively; we can see why if we plot them together with the original sequence Q as in Fig. 8. They form a bounding envelope that encloses Q from above and below. Note that, although the Sakoe-Chiba band is of constant width, the corresponding envelope generally is not of uniform thickness. In particular, the envelope is wider when the underlying query sequence is changing rapidly, and narrower when the query sequence plateaus.

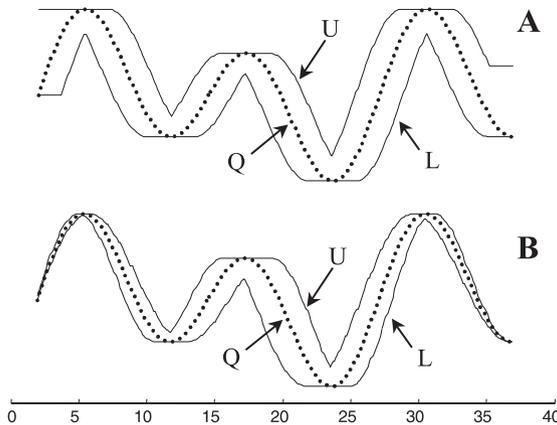


Fig. 8. An illustration of the sequences U and L created for sequence Q (shown dotted). **A** was created using the Sakoe-Chiba band and **B** using the Itakura parallelogram

An obvious but important property of U and L is the following:

$$\forall_i \quad U_i \geq q_i \geq L_i. \tag{8}$$

Having defined U and L , we now use them to define a lower bounding measure for DTW.

$$LB_Keogh(Q,C) = \sqrt{\sum_{i=1}^n \begin{cases} (c_i - U_i)^2 & \text{if } c_i > U_i \\ (c_i - L_i)^2 & \text{if } c_i < L_i \\ 0 & \text{otherwise} \end{cases}} \quad (9)$$

This function can be readily visualized as the Euclidean distance between any part of the candidate matching sequence not falling within the envelope and the nearest (orthogonal) corresponding section of the envelope. Figure 9 illustrates the idea.

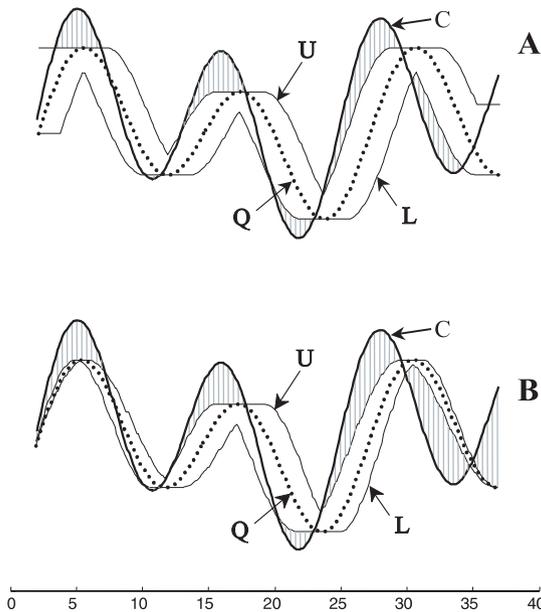


Fig. 9. An illustration of the lower bounding function $LB_Keogh(Q,C)$. The original sequence Q (shown dotted) is enclosed in the bounding envelope of U and L . The squared sum of the distances from every part of the candidate sequence C not falling within the bounding envelope, to the nearest orthogonal edge of the bounding envelope is returned as the lower bound. Bounding envelope **A** was created using the Sakoe-Chiba band and bounding envelope **B** using the Itakura parallelogram

Because the tightness of the bounds is proportional to the number and length of the gray hatch lines, we can see, in this example at least, that the Itakura parallelogram provides a tighter bound than the Sakoe-Chiba band does, and both appear tighter than LB_Kim or LB_Yi in Figs. 4 and 5, respectively.

We will now prove the claim of lower bounding.

Proposition 1. For any two sequences Q and C of the same length n , for any global constraint on the warping path of the form $j-r \leq i \leq j+r$, the following inequality holds: $LB_Keogh(Q,C) \leq DTW(Q,C)$

Proof. We wish to prove

$$\sqrt{\sum_{i=1}^n \begin{cases} (c_i - U_i)^2 & \text{if } c_i > U_i \\ (c_i - L_i)^2 & \text{if } c_i < L_i \\ 0 & \text{otherwise} \end{cases}} \leq \sqrt{\sum_{k=1}^K w_k}.$$

Our strategy will be to assume the opposite and show that it leads to a contradiction. Assume

$$\sqrt{\sum_{i=1}^n \begin{cases} (c_i - U_i)^2 & \text{if } c_i > U_i \\ (c_i - L_i)^2 & \text{if } c_i < L_i \\ 0 & \text{otherwise} \end{cases}} > \sqrt{\sum_{k=1}^K w_k}.$$

Because the terms under the radicals are positive, we can square both sides:

$$\sum_{i=1}^n \begin{cases} (c_i - U_i)^2 & \text{if } c_i > U_i \\ (c_i - L_i)^2 & \text{if } c_i < L_i \\ 0 & \text{otherwise} \end{cases} > \sum_{k=1}^K w_k.$$

From Eq. (3), we know that $n \leq K$ (with $0 \leq K - n \leq n - 2$). So we can match every term on the left-hand side (LHS), with a unique term on the right-hand side (RHS), leaving $K - n$ terms unmatched.

$$\sum_{i=1}^n \begin{cases} (c_i - U_i)^2 & \text{if } c_i > U_i \\ (c_i - L_i)^2 & \text{if } c_i < L_i \\ 0 & \text{otherwise} \end{cases} > \sum_{k \in \text{matched}} w_k + \sum_{k \in \text{unmatched}} w_k.$$

We will map the i^{th} term on the LHS with one of the i^{th} terms on the RHS (recall that w_k is defined as $(i, j)_k$, cf. Sect. 2.1). There may be several values of j for a single i ; so to enforce the desired one-to-one mapping, we will map to the one with the lowest value for j . All the other w_k 's are placed in the unmatched summation.

For the moment, let us ignore the unmatched terms and see what relationship exists between just the matched terms and the LHS. There are three cases to consider; let us consider the case when $c_i > U_i$:

$$\begin{aligned} (c_i - U_i)^2 &<?> w_k \\ (c_i - U_i)^2 &<?> (c_i - q_j)^2 && \text{By definition (cf. Sect. 2.1).} \\ (c_i - U_i) &<?> (c_i - q_j) && \text{Because } c_i > U_i, \text{ we can take square roots.} \\ -U_i &<?> -q_j && \text{Add } -c_i \text{ to both sides.} \\ q_j &<?> U_i && \text{Add } U_i + q_j \text{ to both sides.} \\ q_j &<?> \max(q_{i-r} : q_{i+r}) && \text{By definition, Eq. (6).} \end{aligned}$$

Because we have $n = m$ (recall LB_Keogh is only defined when $|Q| = |C|$), then $j - r \leq i \leq j + r, \Rightarrow i - r \leq j \leq i + r$, so we can rewrite the RHS as

$$q_j <?> \max(q_{i-r}, q_{(i+1)-r}, q_j, \dots, q_{i+r}).$$

If we remove all terms except q_j from the RHS, we are left with

$$q_j \leq \max(q_j).$$

The case when $c_i < L_i$ yields to a similar argument. The third case yields

$$0 \leq (c_i - q_j)^2 \quad \text{Because } (c_i - q_j)^2 \text{ must be nonnegative.}$$

But if all the matched terms in $\sum_{k \in \text{matched}} w_k$ are larger than their matching counterparts on the LHS, then the only hope of our assumption being correct is if $\sum_{k \in \text{unmatched}} w_k$ is a negative number, but the sum of squared terms can never be negative!

Thus, we have shown a contradiction; our assumption was incorrect and $\text{LB_Keogh}(Q,C) \leq \text{DTW}(Q,C)$. \square

In the next section, we will show how LB_Keogh can be indexed.

4. Indexing DTW

Virtually all approaches to indexing time series under the Euclidean distance that guarantee no false dismissals use the GEMINI framework of Faloutsos et al. (Chan et al. 2003; Faloutsos et al. 1994; Keogh et al. 2000, 2001; Korn et al. 1997; Yi and Faloutsos 2000). Using the GEMINI framework, all one has to do is to choose a high level representation of the data and define a lower bounding measure on it (Faloutsos et al. 1994). Many such representations have been suggested, including Fourier transforms (Faloutsos et al. 1994), Wavelets (Chan et al. 2003), singular value decomposition (Korn et al. 1997), adaptive piecewise aggregate approximation (Keogh et al. 2001), and a simple technique independently introduced by two authors called piecewise aggregate approximation (PAA) (Keogh et al. 2000; Yi and Faloutsos 2000). This technique is attractive because it is simple, intuitive, and competitive with the other more complex approaches. In this section, we will show that PAA can be adapted to allow indexing under DTW. We begin with a brief review of PAA.

4.1. Piecewise aggregate approximation

We have previously denoted a time series as $C = c_1, \dots, c_n$. We assume each sequence in our database is n units long. Let N be the dimensionality of the space we wish to index ($1 \leq N \leq n$). For convenience, we assume that N is a factor of n . While this is not a requirement of our approach, it does simplify the notation.

A time series C of length n can be represented in N -dimensional space by a vector $\bar{C} = \bar{c}_1, \dots, \bar{c}_N$. The i^{th} element of \bar{C} is calculated by the following equation:

$$\bar{c}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} c_j. \quad (10)$$

Simply stated, to reduce the time series from n dimensions to N dimensions, the data is divided into N equal-sized frames. The mean value of the data falling within a frame is calculated, and a vector of these values becomes the data-reduced representation. The complicated subscripting in Eq. (10) just insures that the original sequence is divided into the correct number and size of frames. The representation can best be visualized as an attempt to model the original time series with a linear combination of box basis functions as shown in Fig. 10.

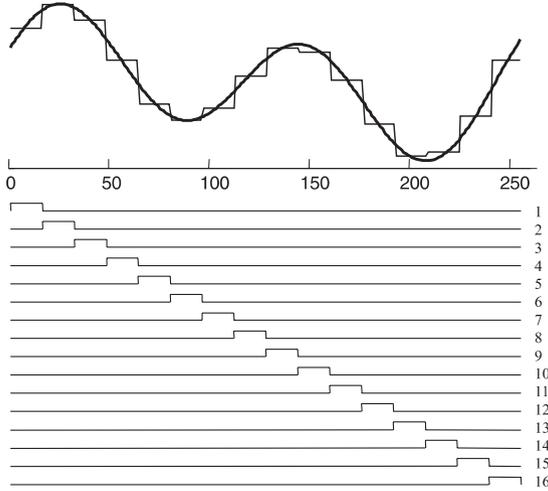


Fig. 10. The PAA representation can be readily visualized as an attempt to model a sequence with a linear combination of box basis functions. In this case, a sequence of length 256 is reduced to 16 dimensions

Given two original sequences Q and C , we can transform them into \bar{Q} and \bar{C} using Eq. (10), and approximate their Euclidean distance by:

$$DR(\bar{Q}, \bar{C}) \equiv \sqrt{\frac{n}{N}} \sqrt{\sum_{i=1}^N (\bar{q}_i - \bar{c}_i)^2}. \quad (11)$$

A proof that $DR(\bar{Q}, \bar{C})$ lower bounds the true Euclidean distance is in Keogh et al. (2000) (a different proof appears in Yi and Faloutsos (2000)).

4.2. Modifying PAA to index time-warped queries

In Sect. 3, we introduced the lowering bounding function `LB_Keogh`; However, calculating this function requires n values. Because n may be in the order of hundreds to thousands and multidimensional index structures begin to degrade rapidly somewhere above 16 dimensions (Hellerstein et al. 1997; Seidl and Kriegel 1988), we need a way to create a lower, N -dimension version of the function, where N is a number that can be reasonably handled by a multidimensional index structure (Guttman 1984). We also need this lower dimension version of the function to lower bound `LB_Keogh` (and therefore, by transitivity, DTW).

We begin by creating special piecewise aggregate approximations of U and L , which we will denote as \hat{U} and \hat{L} . Although they are piecewise aggregate approximations, the definitions of \hat{U} and \hat{L} differ from those we have seen in Eq. (10); in particular, we have

$$\hat{U}_i = \max \left(U_{\frac{n}{N}(i-1)+1}, \dots, U_{\frac{n}{N}i} \right) \quad (12)$$

$$\hat{L}_i = \min \left(L_{\frac{n}{N}(i-1)+1}, \dots, L_{\frac{n}{N}i} \right). \quad (13)$$

We can visualize \hat{U} and \hat{L} as the piecewise constant functions that bound, without intersecting, U and L, respectively. Figure 11 illustrates this intuition.

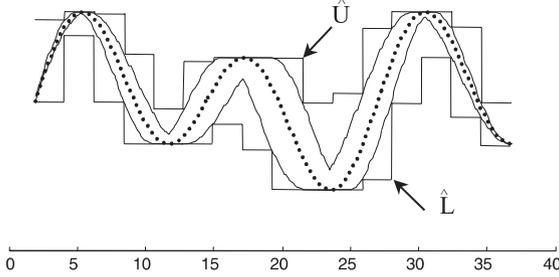


Fig. 11. We can readily visualize \hat{U} and \hat{L} as the piecewise constant functions that bound, without intersecting, U and L, respectively

We are now able to define the low dimension, lower bounding function, which we denote as LB_PAA. Given a candidate sequence C, transformed to \bar{C} by Eq. (10), and a query sequence Q, with its companion PAA functions \hat{U} and \hat{L} , the following function lower bounds LB_Keogh:

$$LB_PAA(Q, \bar{C}) = \sqrt{\sum_{i=1}^N \frac{n}{N} \begin{cases} (\bar{c}_i - \hat{U}_i)^2 & \text{if } \bar{c}_i > \hat{U}_i \\ (\bar{c}_i - \hat{L}_i)^2 & \text{if } \bar{c}_i < \hat{L}_i \\ 0 & \text{otherwise} \end{cases}} \quad (14)$$

The proof that $LB_PAA(Q, \bar{C}) \leq LB_Keogh(Q, C)$ is a straightforward but long extension of Proposition 1; we omit it for brevity.

The final step necessary to allow indexing is to define a MINDIST(Q,R) function that returns a lower bounding measure of the distance between a query Q and R, where R is a minimum bounding rectangle (MBR).

Suppose our index structure contains a leaf node U. Let R = (L,H) be the MBR associated with U, where L = $\{l_1, l_2, \dots, l_N\}$ and H = $\{h_1, h_2, \dots, h_N\}$ are the lower and higher endpoints of the major diagonal of R. By definition, R is the smallest rectangle that spatially contains each PAA point $\bar{C} = \bar{c}_1, \dots, \bar{c}_N$ stored in U. Given the above, MINDIST(Q,R) is defined as

$$MINDIST(Q, R) = \sqrt{\sum_{i=1}^N \frac{n}{N} \begin{cases} (l_i - \hat{U}_i)^2 & \text{if } l_i > \hat{U}_i \\ (h_i - \hat{L}_i)^2 & \text{if } h_i < \hat{L}_i \\ 0 & \text{otherwise} \end{cases}} \quad (15)$$

This function is visualized in Fig. 12.

Having defined LB_PAA and MINDIST(Q,R), we are now ready to introduce the K-nearest neighbor search (K-NN) algorithm. The basic algorithm is shown in Table 3. It is an optimization on the GEMINI K-NN algorithm (Faloutsos et al. 1994) as suggested by Seidl and Kriegel (1988) and is a modification of the algorithm used for indexing time series under the Euclidean metric in Keogh et al. (2001).

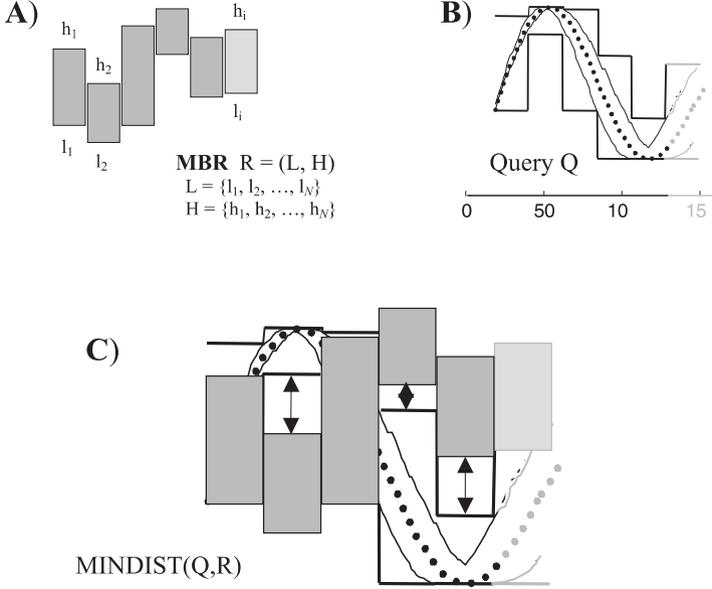


Fig. 12. **A)** A representation of a minimum bounding rectangle (MBR). **B)** A subsection of the query shown in Fig. 11, with its attendant functions \hat{U} and \hat{L} . **C)** An illustration of the MINDIST function. The lengths of the arrow lines, squared, summed and square rooted, are returned as the minimum distance between Q and any sequence contained within R

A query $KNNSearch(Q, K)$ with query sequence Q and desired number of neighbors K retrieves a set \mathcal{C} of K time series such that, for any two Sequences, $C \in \mathcal{C}$, $E \notin \mathcal{C}$, and $DTW(Q, C) \leq DTW(Q, E)$. Like the classic K -NN algorithm (Roussopoulos et al. 1995), the algorithm in Table 3 uses a priority *queue* to visit nodes/objects in the index in the increasing order of their distances from Q in the indexed (i.e. PAA) space. The distance of an object (i.e. PAA point) C from Q is defined by $LB_PAA(Q, \bar{C})$ (cf. Sect. 4.2, Eq. (14)) while the distance of a node U from Q is defined by the minimum distance $MINDIST(Q, R)$ of the minimum bounding rectangle (MBR) R associated with U from Q .

We begin by pushing the root node of the index into the *queue* (line 1). The algorithm navigates the index by popping out the item from the top of the queue at each step (line 8). If the popped item is a PAA point C , we go to disk to retrieve the original time series C , and we compute its exact distance $DWT(Q, C)$ from the query and then insert it into a temporary list *temp* (lines 9–11). If, on the other hand, the popped item is a node of the index structure, we compute the distance of each of its children from Q and push them into *queue* (Lines 12–17).

We only move a sequence C from *temp* to *result* when we are sure that it is one of the K -NN of Q . That is to say, there exists no object $E \notin result$ such that $DTW(Q, E) < DTW(Q, C)$ and $|result| < K$. This second condition is guaranteed by the exit condition in line 7. The first condition can be guaranteed as follows. Let \mathbf{I} be the set of PAA points retrieved thus far using the index (i.e. $\mathbf{I} = temp \cup result$). If we can guarantee that $\forall C \in \mathbf{I}, \forall E \notin \mathbf{I}, LB_PAA(Q, \bar{C}) \leq DTW(Q, E)$, then the condition “ $DTW(Q, C) \leq top.dist$ ” in line 4 will ensure that there exists no unexplored sequence E such that $DTW(Q, E) < DTW(Q, C)$.

Table 3. K-NN algorithm to compute the exact K nearest neighbors of a query time series Q using a multi-dimensional index structure

Algorithm KNNSearch(Q, K)
<pre> Variable queue: MinPriorityQueue; Variable list: temp; 1. queue.push(root_node_of_index, 0); 2. while not queue.IsEmpty() do 3. top = queue.Top(); 4. for each time series C in temp such that $DTW(Q, C) \leq top.dist$ 5. Remove C from temp; 6. Add C to result; 7. if result = K return result; 8. queue.Pop(); 9. if top is a PAA point C 10. Retrieve full sequence C from database; 11. temp.insert(C, $DTW(Q, C)$); 12. else if top is a leaf node 13. for each data item C in top 14. queue.push(C, $LB_PAA(Q, \bar{C})$); 15. else // top is a non-leaf node 16. for each child node U in top 17. queue.push(U, $MINDIST(Q, R)$) // R is MBR associated with U. </pre>

By inserting the time series in *temp* (i.e. previously seen objects) into *result* in increasing order of their distances $DTW(Q, C)$ (by keeping *temp* sorted by $DTW(Q, C)$), we ensure that there exists no explored object E such that $DTW(Q, E) < DTW(Q, C)$.

The definitions of LB_Keogh, LB_PAA, and MINDIST proposed in this work are also needed for answering range queries using a multidimensional index structure. We can use a classic R-tree-style recursive search algorithm. Because both $MINDIST(Q, R)$ and $LB_PAA(Q, \bar{C})$ lower bound $DTW(Q, C)$, the algorithm shown in Table 4 is correct (Faloutsos and Lin 1995).

Table 4. Range search algorithm to retrieve all the time series within a range of ϵ from query time series Q. The function is invoked as RangeSearch(Q, ϵ , root_node_of_index)

Algorithm RangeSearch(Q, ϵ , T)
<pre> 1. if T is a non-leaf node 2. for each child U of T 3. if $MINDIST(Q, R) \leq \epsilon$ RangeSearch(Q, ϵ, U); // R is MBR of U 4. else // T is a leaf node 5. for each PAA point C in T 6. if $LB_PAA(Q, \bar{C}) \leq \epsilon$ 7. Retrieve full sequence C from database; 8. if $DTW(Q, C) \leq \epsilon$ Add C to result; </pre>

5. Experimental evaluation

In this section, we test our proposed approach with a comprehensive set of experiments.

5.1. Experimental philosophy

Previous experience in reimplementing and testing more than a dozen different Euclidean time series indexing techniques (Keogh et al. 2000, 2001), suggests that many published results do not generalize to real-world datasets and conditions. We therefore conducted the experiments in this paper with the explicit goal of conducting the most comprehensive and detailed set of time series indexing experiments ever attempted. In particular, we have taken the following steps to insure the most meaningful and generalizable results.

- Instead of testing on just one or two datasets, as is typical (Agrawal et al. 1995; Berndt and Clifford 1994; Chan et al. 2003; Faloutsos et al. 1994; Kim et al. 2001; Park et al. 1999, 2000, 2001), we tested all algorithms on 32 datasets. These datasets cover the complete spectrum of stationary/nonstationary, noisy/smooth, cyclical/noncyclical, symmetric/asymmetric, etc. The data also represents the many areas in which DTW is used, including finance, medicine, biometrics, chemistry, astronomy, robotics, networking, and industry.
- We designed our experiments to be completely reproducible. We saved every random number, every setting and all data, and have made them available on a free CD-ROM.
- To ensure true randomness where required, we used random numbers created by a quantum mechanical process (Walker 2001).
- Although we also present results of an implemented system, we present comprehensive results that are completely independent of implementation details (i.e. page size, cache size, etc). This is to guard against implementation bias (Hellerstein et al. 1997; Keogh et al. 2001) and to allow and encourage independent replication of our results.

For simplicity and brevity, we only show results for nearest neighbor queries; however, we obtained very similar results for range queries. Because of the large volume of experiments conducted, in this section, we will present graphics to summarize our findings and we will reproduce the actual numbers in Appendix A.

Unless otherwise stated, we used the Sakoe-Chiba band with a width of 10% of n , because this appears to be the most commonly used constraint in the literature (Rabiner et al. 1978; Sakoe and Chiba 1978). We note that, had we used the Itakura parallelogram instead, our results would be even better (depending on the dataset, by an approximate factor of 4–12).

5.2. Comparison of lower bounding functions

We begin our experiments with a comparison of the tightness of the lower bounds for the three functions LB_Yi, LB_Kim, and LB_Keogh. We define T as the ratio of the estimated distance between two sequences over the true distance between the same two sequences.

$$T = \frac{\text{Lower Bound Estimate of Dynamic Time Warp Distance}}{\text{True Dynamic Time Warp Distance}} \quad (16)$$

T is in the range $[0, 1]$, with the larger the better. To estimate T for each of the 32 datasets, we did the following: We randomly extracted 50 sequences of length 256. We compared each sequence to the 49 others, using the true DTW distance, and the three lower bounding functions. For each dataset, we report T as the average ratio from the 1,225 ($50 \cdot 49/2$) comparisons made.

Figure 13 shows the results of the experiments. On 24 out of 32 datasets, LB_Yi produces tighter bounds than LB_Kim, and its average value is approximately 1.38 times larger. The most obvious result from the experiment, however, is the dominance of LB_Keogh. It wins on every dataset, and its average value is approximately 3.11 times larger than its nearest rival. Because the efficiency of indexing has a (much) greater than linear dependence on the tightness of the lower bounding function, these results augur well for our approach.

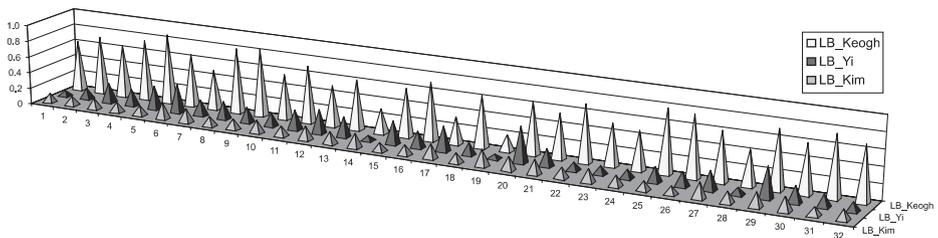


Fig. 13. The mean value of T (tightness of lower bound) for the three lower bounding functions under consideration for 32 datasets from finance, medicine, biometrics, chemistry, physics, astronomy, robotics, networking, and industry. Appendix A contains a key to the datasets

We choose to report results from a query length of 256 because this is about the midrange of queries reported in the literature (Chan et al. 2003; Chu et al 2002; Park et al. 1999; Yi et al. 1998). However, we also experimented with queries in the range of 32-1,024. This range was chosen to include the longest and shortest reported in the literature (Chan et al. 2003; Park, personal communication). All techniques perform better for short queries; however, while both LB_Kim and LB_Yi degrade rapidly for longer queries, LB_Keogh stays almost constant for longer queries. This effect was observed on all datasets. For brevity, we just present results for the random walk dataset in Fig. 14.

5.3. Comparison of pruning power

To compare the pruning power of the three techniques under consideration, we measure P , the fraction of the database that does *not* require full computation of DTW while still allowing us to guarantee that we have found the nearest match to a 1-NN query.

$$P = \frac{\text{Number of objects that do not require full DTW}}{\text{Number of objects in database}} \quad (17)$$

To calculate P , we do the following. From each of the 32 datasets, we randomly extract 50 sequences of length 256. For each of the 50 sequences, we separate out the sequence from the other 49 sequences. We then find the nearest match to our withheld sequence among the remaining 49 sequences using the sequential scan algorithm of Table 2. We measure the number of times we can use the linear-time

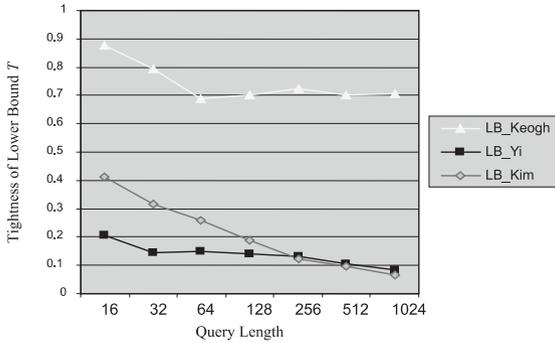


Fig. 14. The effect of query length on the tightness of lower bounds for the three techniques under consideration

lower bounding functions to prune away the quadratic-time computation of the full DTW algorithm. For fairness, we visit the 49 sequences in the same order for each approach. The value P reported is averaged over all 50 runs.

Note the value of P depends only on the data and is completely independent of any implementation choices, including spatial access method, buffer size, computer language, or hardware platform. A similar idea for evaluating indexing schemes appears in Hellerstein et al. (1997).

The results are summarized in Fig. 15. On 25 out of 32 datasets, LB_Yi is more efficient at pruning than LB_Kim. On average, it was able to prune 1.53 times as many items. Once again, however, the most obvious result is the dominance of LB_Keogh. It wins on every dataset and was able to prune 3.95 times as many items as LB_Yi and 6.06 times as many items as LB_Kim.

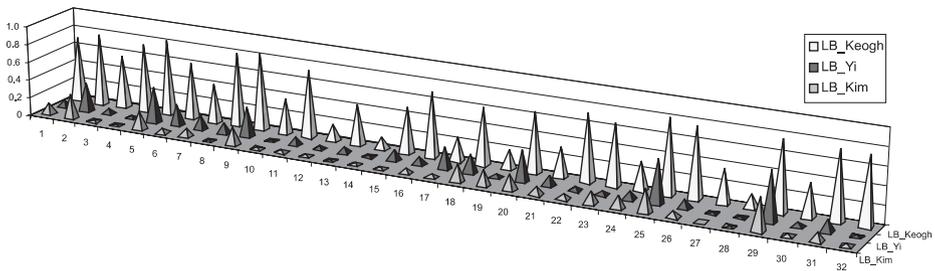


Fig. 15. The mean value of P (pruning power) for the three lower bounding functions under consideration for 32 datasets from finance, medicine, biometrics, chemistry, physics, astronomy, robotics, networking, and industry. Appendix A contains a key to the datasets

Note that, while these results are powerful implementation-independent predictors of indexing performance, they may actually be pessimistic. There are two related reasons why. First, the sequential scan algorithm of Table 2 is inefficient, as it visits the items and calculates the DTW measures (where necessary) in a predefined order. A more efficient implementation would sort and then visit the sequences, in ascending order of the lower bounding distance. This, of course, is essentially what spatial indexing does.

The second reason why the results may be pessimistic predictors of indexing performance is the relatively small size of the datasets. We should expect the fraction of pruned sequences to increase on larger datasets. The reason is because the larger the dataset, the greater the chance there is of a good match being found, and a good match allows us to extract the maximum benefit from the pruning conditional $LB_dist < best_so_far$ in line 4 of the algorithm in Table 2. To demonstrate this effect, we ran the same experiment above on increasingly larger subsets of the random walk dataset. The results are shown in Fig. 16.

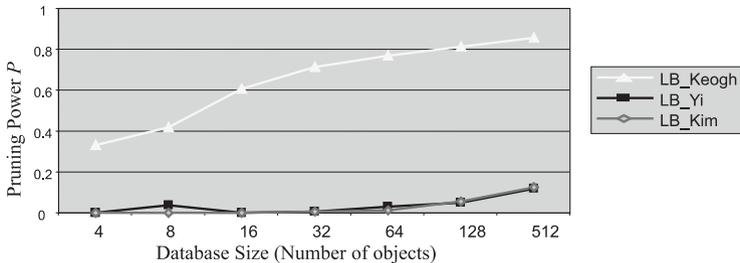


Fig. 16. The effect of database size on pruning power. Note that, as the size of the database increases, we are able to prune a larger fraction of the data

5.4. Experiments on an implemented system

The 32 datasets used in the previous experiments illustrate the dominance of the proposed approach on a wide variety of datasets. However, most are not large enough by themselves to warrant the title of large database. We therefore pooled all 32 datasets into a single dataset that we call mixed bag (MB). In addition to this ultraheterogeneous data, we created a very large database of random walk data (RW II) because this is the most studied dataset for indexing comparisons (Chan et al. 2003; Chu et al. 2002; Keogh et al 2000, 2001; Park et al. 1999, 2001; Yi et al. 1998), and is, by contrast with the above, a very homogeneous dataset. Details of these datasets appear in Appendix A.

We performed experiments on AMD Athlon 1.4 GHz processor, with 512 MB of physical memory and 57.2 GB of secondary storage. The spatial access method used was the R-tree (Gollmer and Posten 1995).

To evaluate the performance of the proposed technique we used the normalized CPU cost.

Definition. *The Normalized CPU cost:* The ratio of average CPU time to execute a query using the index to the average CPU time required performing a linear (sequential) scan. The normalized cost of a linear scan is 1.0.

Beating a linear scan is nontrivial because it can take advantage of sequential disk access, whereas any indexing technique must make random disk accesses. It is generally understood that random access is about ten times slower than sequential access (Hellerstein et al. 1997; Roussopoulos et al. 1995; Seidl and Kriegel 1988). For fairness, we allowed linear scan to utilize the lower bounding function LB_Keogh.

Because there is no known exact indexing method for LB_Yi, we could not include it in this experiment. We originally included LB_Kim in the experiments, but

found that it never beat the linear scan, and we therefore decided to exclude it from graphic presentation.

We tested over a range of query lengths and dimensionalities, but show just one typical result for brevity. Figure 17 shows the normalized CPU cost of linear scan and LB_Keogh, for queries of length 256, with a 16-dimensional index, for increasingly large databases.

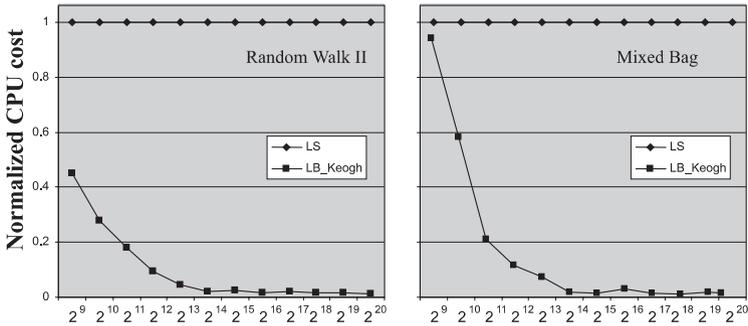


Fig. 17. The normalized CPU cost of linear scan and LB_Keogh, for queries of length 256, with a 16-dimensional index, for increasingly large databases. Note that the X-axis is in logarithmic scale and denotes the number of items in the database

5.5. The effect of changing the warping window width on efficiency

The experiments above convincingly demonstrate the superiority of the proposed lower bounding technique for different datasets, different query lengths, different database sizes, etc. However, all the experiments use a Sakoe-Chiba band with a width of 10% of the query length because, as noted above, this seems to be the most common constraint used in practice (Rabiner et al. 1978; Sakoe and Chiba 1978). However, it is natural to ask how sensitive the results are to this parameter. To find out, we repeated the experiment in Sect. 5.2, this time also testing a warping window twice as wide and half as wide as the original experiments. Because showing the results for the entire 32 datasets is not practical, we chose to report only the first, middle and last datasets from the list in Appendix A. The results are shown in Fig. 18.

The results are excellent. Even with an extremely wide warping window, we still convincingly beat the two competing approaches. Furthermore, tightening the warping window to a still realistic value of 5% of the query length produces an extraordinary tight lower bound.

5.6. The effect of changing the warping window width on accuracy

In Sect. 3.3.1, we justified using warping windows by noting that researchers who use DTW to solve real-world problems have documented their utility (Aach and Church 2001; Caiani et al. 1998; Gavrila and Davis 1995; Gollmer and Posten 1995; Itakura 1975; Kovacs-Vajna 2000; Munich and Perona 1999; Rath and Manmatha 2002). However, because warping windows are the cornerstone of our lower bounding technique, we will conduct experiments to explicitly justify their use. As we are

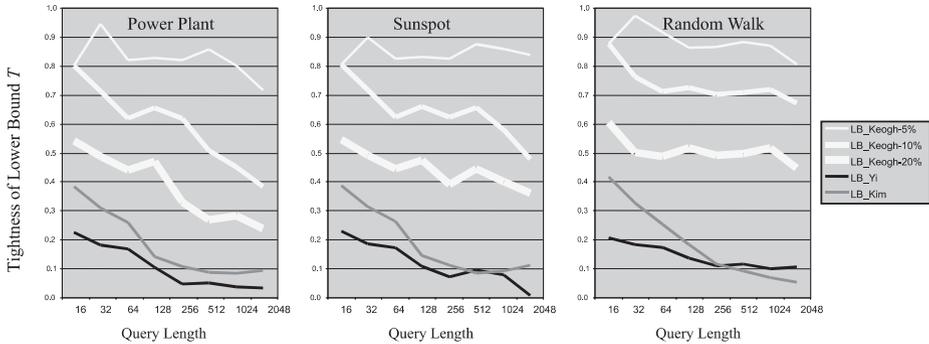


Fig. 18. The effect of warping window width on the tightness of lower bounds for various query lengths. Note that, even with an extremely loose warping window, equal to 20% of the query length, the lower bounds for LB_Keogh are tighter than the two competing approaches over the entire range of query lengths

interested in indexing data, an appropriate way to do this might be to measure precision/recall under varying warping windows. However, to our knowledge, there are no large time series datasets that have been annotated as relevant/irrelevant for given queries. Instead, we will consider the effect of warping windows on classification of time series because accuracy in classification is a close analogue of precision/recall in information retrieval.

To reduce the possibility of data bias, we will only test datasets that have appeared in the literature several times and are publicly available (Kadous 1999; Diez and Gonzalez 2000). In particular, we tested the following datasets, which are illustrated in Fig. 19A.

- **Transient Classification Benchmark (TCB):** A synthetic dataset designed to simulate instrumentation failures in a nuclear power plant. This is a multiclass, multidimensional dataset. For simplicity, we use only sensor 3 of classes 3 and 6. There are 50 instances of length 275 in each class (data courtesy of Davide Rovero).
- **Australian Sign Language (ASL):** This dataset consists of the X-axis motion of a subject’s right hand as they sign the words *girl*, *mine*, *read* and *thank* in Australian Sign Language (Kadous 1999). This is a nontrivial classification task because the data is undersampled (only 30 points long) and very noisy. In addition, of the 20 instances in each class, each was created by four different individuals on five different days.

For each dataset, we evaluated a one nearest neighbor classifier for every possible width of the Sakoe-Chiba band from 1 to n . Each classifier was evaluated using the leaving-one-out cross validation. The results are shown in Fig. 19B.

We can make the following observations about the experiments. It is yet another confirmation of the superiority of DTW over Euclidean distance (recall that warping window width = 1, is equivalent to Euclidean distance). A more interesting observation is that, while some DTW helps, there are diminishing returns. In the case of ASL, too much freedom in warping actually hurts the accuracy. These results strongly support the idea that constraining DTW with warping windows is a good idea, independent of our ability to exploit it for speedup. The results also suggest an interesting research direction; can one automatically learn the best warping window for a particular dataset and query? We are actively exploring this question.

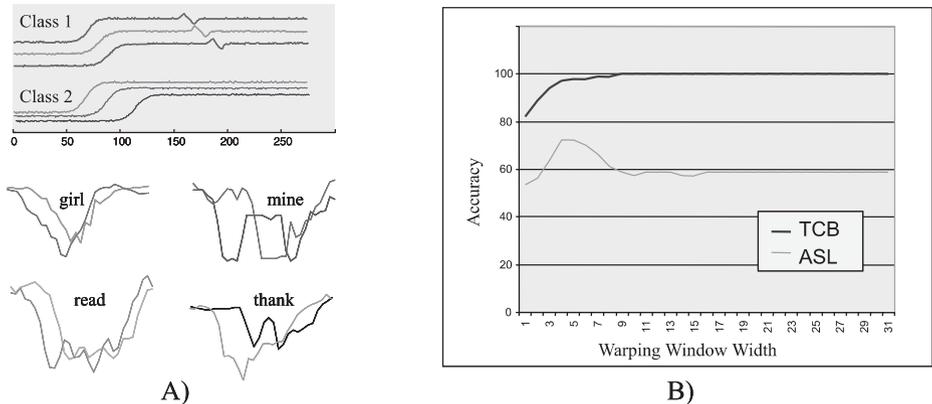


Fig. 19. A) *Top*: examples of the two classes from the TCB dataset. *Bottom*: Examples of the four classes from the ASL dataset. B) The effect of varying the warping window width on accuracy for the two datasets in question

6. Discussion and conclusions

In one of the most referenced papers on time series similarity ever published (Agrawal et al. 1995), the authors explicitly state, “Dynamic time warping ... cannot be speeded up by indexing.” This sentiment has since been echoed in several dozen other papers (Chan et al. 2003; Yi et al. 1998). How then have we achieved the seemingly impossible? First, we have only considered the case where the two sequences are of the same length. This is not really a limitation because the user can always reinterpolate the query to any desired length in $O(n)$ time. Second, we can only index sequences if we assume the warping path is constrained. Once again, we feel that this is not really a restriction because virtually every practitioner we are aware of reiterates the absolute necessity of using constraints (Aach and Church 2001; Berndt and Clifford 1994; Caiani et al. 1998; Chu et al. 2002; Gollmer and Posten 1995; Itakura 1975; Rabiner et al. 1978; Sakoe and Chiba 1978; Schmill et al. 1999; Strik and Boves 1988; Tappert and Das 1978).

Our approach is particularly attractive because, as a special case (r is set to zero), it degenerates to Euclidean indexing using PAA, an approach that has been shown by two independent groups of researchers to be state of the art in terms of efficiency and flexibility (Chu et al. 2002; Keogh et al. 2000; Yi et al. 1998).

There are several directions in which this work may be extended. For example, we note that some algorithms for matching two and three-dimensional shapes are very close analogues of the DTW algorithm and thus may benefit from a similar lower bounding function. In addition, Rath and Manmatha (Rath and Manmatha 2002) have informed us that they have generalized LB_Keogh to multidimensional time series and intend to use the resulting algorithm for indexing massive repositories of handwritten historical documents (personal communication).

Acknowledgements. Thanks to the anonymous reviewers, Kaushik Chakrabarti, Dennis De-Coste, Sharad Mehrotra, Per-Åke (Paul) Larson and Michalis Vlachos for their useful comments on a preliminary version of this paper. Thanks also to the many donors of the data used in this work.

Appendix A

The raw numbers obtained from the experiments discussed in Sects. 5.2 and 5.3 are shown in Table 5. These numbers may be visualized in Figs. 13 and 15, respectively.

Table 5. The raw numbers obtained from the experiments discussed in Sects. 5.2 and 5.3

ID	Name	Size	<i>T</i> (Tightness of Lower Bound)			<i>P</i> (Pruning Power)		
			LB_Kim	LB_Yi	LB_Keogh	LB_Kim	LB_Yi	LB_Keogh
1	Sunspot	2,899	0.11	0.06	0.63	0.14	0.07	0.73
2	Power	35,040	0.12	0.13	0.73	0.27	0.32	0.80
3	ERP data	198,400	0.13	0.24	0.65	0.01	0.07	0.59
4	Spot exrates	2,567	0.12	0.21	0.75	0.01	0.03	0.77
5	Shuttle	6,000	0.12	0.29	0.87	0.20	0.39	0.85
6	Water	6,573	0.22	0.36	0.66	0.05	0.24	0.64
7	Chaotic	1,800	0.18	0.19	0.50	0.09	0.16	0.43
8	Steamgen	38,400	0.11	0.22	0.81	0.00	0.11	0.82
9	Ocean	4,096	0.13	0.19	0.84	0.20	0.34	0.87
10	Tide	8,746	0.16	0.16	0.56	0.02	0.01	0.39
11	CSTR	22,500	0.13	0.25	0.71	0.04	0.09	0.75
12	Winding	17,500	0.17	0.29	0.51	0.03	0.04	0.19
13	Dryer2	5,202	0.15	0.25	0.62	0.01	0.07	0.46
14	Robot arm	2,048	0.18	0.06	0.30	0.03	0.01	0.13
15	Ph Data	6,003	0.11	0.29	0.60	0.03	0.12	0.51
16	Power Plant	2,400	0.13	0.20	0.72	0.05	0.08	0.72
17	Evaporator	37,830	0.18	0.31	0.34	0.04	0.25	0.26
18	Ballbeam	2,000	0.12	0.15	0.65	0.20	0.21	0.63
19	Tongue	700	0.20	0.06	0.21	0.17	0.04	0.21
20	Fetal ECG	22,500	0.17	0.45	0.66	0.17	0.35	0.67
21	Balloon	4,002	0.18	0.22	0.55	0.09	0.12	0.33
22	Stand' & Poor	17,610	0.13	0.10	0.71	0.06	0.04	0.75
23	Speech	1,020	0.16	0.11	0.53	0.14	0.04	0.69
24	Soil temp	2,304	0.14	0.11	0.48	0.13	0.08	0.32
25	Wool	2,790	0.11	0.19	0.79	0.26	0.48	0.83
26	Infrasound	8,192	0.10	0.18	0.76	0.07	0.09	0.78
27	Network	18,000	0.14	0.18	0.55	0.00	0.01	0.38
28	EEG	11,264	0.14	0.08	0.44	0.01	0.01	0.16
29	Koski ECG	144,002	0.18	0.39	0.73	0.36	0.54	0.78
30	Buoy sensor	55,964	0.17	0.20	0.61	0.03	0.06	0.38
31	Burst	9,382	0.10	0.15	0.77	0.09	0.12	0.76
32	Random walk I	65,536	0.13	0.11	0.68	0.03	0.02	0.75
		Mean Value	0.144	0.199	0.622	0.094	0.144	0.572
MB	Mixed bag	763,270						
RW	Random walk II	1,048,576						

References

- Aach J, Church G (2001) Aligning gene expression time series with time warping algorithms. *Bioinformatics* 17:495–508
- Agrawal R, Lin KI, Sawhney HS, Shim K (1995) Fast similarity search in the presence of noise, scaling, and translation in times-series databases. In: *Proceedings of the 21st international conference on very large databases*, pp 490–501
- Bar-Joseph Z, Gerber G, Gifford D, Jaakkola T, Simon I (2002) A new approach to analyzing gene expression time series data. In: *Proceedings of the 6th annual international conference on research in computational molecular biology*, pp 39–48
- Berndt D, Clifford J (1994) Using dynamic time warping to find patterns in time series. *AAAI-94 workshop on knowledge discovery in databases*, pp 229–248
- Caiani EG, Porta A, Baselli G, Turiel M, Muzzupappa S, Pieruzzi F, Crema C, Malliani A, Cerutti S (1998) Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume. *IEEE Comput Cardiol* 25:73–76
- Chan KP, Fu A, Yu C (2003) Haar wavelets for efficient similarity search of time-series: with and without time warping. *IEEE Trans Knowl Data Eng* 15(3):686–705
- Chu S, Keogh E, Hart D, Pazzani M (2002) Iterative deepening dynamic time warping for time series. In: *Proceedings of the 2nd SIAM international conference on data mining*
- Das G, Lin K, Mannila H, Renganathan G, Smyth P (1998) Rule discovery from time series. *Proceedings of the 4th international conference of knowledge discovery and data mining*. AAAI Press, pp 16–22
- Debregeas A, Hebrail G (1998) Interactive interpretation of Kohonen maps applied to curves. *Proceedings of the 4th international conference of knowledge discovery and data mining*, pp 179–183
- Diez JJR, Gonzalez CA (2000) Applying boosting to similarity literals for time series classification. *Multiple classifier systems, 1st international workshop*, pp 210–219
- Faloutsos C, Ranganathan M, Manolopoulos Y (1994) Fast subsequence matching in time-series databases. In: *Proceedings of the ACM SIGMOD conference*, Minneapolis, MN, pp 419–429
- Faloutsos C, Lin K (1995) FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *SIGMOD conference*, pp 163–174
- Gavrila DM, Davis LS (1995) Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In: *International workshop on automatic face- and gesture-recognition*, pp 272–277
- Gollmer K, Posten C (1995) Detection of distorted pattern using dynamic time warping algorithm and application for supervision of bioprocesses. *On-line fault detection and supervision in chemical process industries*
- Guttman A (1984) R-trees: A dynamic index structure for spatial searching. In: *Proceedings ACM SIGMOD conference*, pp 47–57
- Hellerstein JM, Papadimitriou CH, Koutsoupias E (1997) Towards an analysis of indexing schemes. *16th ACM symposium on principles of database systems*, pp 249–256
- Itakura F (1975) Minimum prediction residual principle applied to speech recognition. *IEEE Trans Acoustics Speech Signal Process ASSP* 23:52–72
- Kadous MW (1999) Learning comprehensible descriptions of multivariate time series. In: *Proceedings of the 16th international machine learning conference*, pp 454–463
- Keogh E, Chakrabarti K, Pazzani M, Mehrotra S (2000) Dimensionality reduction for fast similarity search in large time series databases. *J Knowl Inf Syst* 3(3):263–286
- Keogh E, Chakrabarti K, Pazzani M, Mehrotra S (2001) Locally adaptive dimensionality reduction for indexing large time series databases. In: *Proceedings of ACM SIGMOD conference on management of data*, May, pp 151–162
- Keogh E, Pazzani M (2000) Scaling up dynamic time warping for data mining applications. In: *6th ACM SIGKDD international conference on knowledge discovery and data mining*, Boston
- Kim S, Park S, Chu W (2001) An index-based approach for similarity search supporting time warping in large sequence databases. In: *Proceedings of the 17th international conference on data engineering*, pp 607–614
- Kollios G, Vlachos M, Gunopulos G (2002) Discovering similar multidimensional trajectories. In: *Proceedings of the 18th international conference on data engineering*
- Korn F, Jagadish H, Faloutsos C (1997) Efficiently supporting ad hoc queries in large datasets of time sequences. In: *Proceedings of SIGMOD '97*, pp 289–300
- Kovacs-Vajna ZM (2000) A fingerprint verification system based on triangular matching and dynamic time warping. *IEEE Trans Pattern Anal Mach Intell* 22(11):1266–1276
- Kruskall JB, Liberman M (1983) The symmetric time warping algorithm: from continuous to discrete. In: *Time warps, string edits and macromolecules*. Addison

- Kwong S, He Q, Man K (1996) Genetic time warping for isolated word recognition. *Int J Patt Recogn Artif Intell* 10(7):849–865
- Munich M, Perona P (1999) Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In: *Proceedings of 7th international conference on computer vision, Korfu, Greece*, pp 108–115
- Myers C, Rabiner L, Roseneberg A (1980) Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Trans Acoustics Speech Signal Process ASSP-28*:623–635
- Park S, Lee D, Chu W (1999) Fast retrieval of similar subsequences in long sequence databases. In: *3rd IEEE knowledge and data engineering exchange workshop*
- Park S, Kim S, Chu W (2001) Segment-based approach for subsequence searches in sequence databases. In: *Proceedings of the 16th ACM symposium on applied computing, Las Vegas, NV*, pp 248–252
- Park S, Chu W, Yoon J, Hsu C (2000) Efficient searches for similar subsequences of different lengths in sequence databases. In: *Proceedings of the 16th IEEE international conference on data engineering*, pp 23–32
- Rabiner L, Juang B (1993) *Fundamentals of speech recognition*. Prentice, Englewood Cliffs, NJ
- Rabiner L, Rosenberg A, Levinson S (1978) Considerations in dynamic time warping algorithms for discrete word recognition. *IEEE Trans Acoustics Speech Signal Process ASSP-26*:575–582
- Rath T, Manmatha R (2002) Word image matching using dynamic time warping. *Tec Report MM-38*. Center for Intelligent Information Retrieval, University of Massachusetts Amherst
- Roussopoulos N, Kelley S, Vincent F (1995) Nearest neighbor queries. *SIGMOD Conference*, pp 71–79
- Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans Acoustics Speech Signal Process ASSP 26*:43–49
- Schmill M, Oates T, Cohen P (1999) Learned models for continuous planning. In: *7th international workshop on artificial intelligence and statistics*
- Seidl T, Kriegel H (1998) Optimal multi-step k-nearest neighbor search. *SIGMOD Conference*, pp 154–165
- Strik H, Boves L (1988) Averaging physiological signals with the use of a DTW algorithm. In: *Proceedings SPEECH'88, 7th FASE symposium, Edinburgh, Book 3*, pp 883–890
- Tappert C, Das S (1978) Memory and time improvements in a dynamic programming algorithm for matching speech patterns. *IEEE Trans Acoustics Speech Signal Process ASSP 26*:583–586
- Walker J (2001) HotBits: genuine random numbers generated by radioactive decay, www.fourmilab.ch/hotbits/
- Yi B, Jagadish K, Faloutsos H (1998) Efficient retrieval of similar time sequences under time warping. In: *ICDE 98*, pp 23–27
- Yi BK, Faloutsos C (2000) Fast time sequence indexing for arbitrary L_p norms. *Proceedings of the 26th international conference on very large databases*, pp 385–394

Author biographies



Eamonn Keogh is an assistant professor of computer science at the University of California, Riverside. His research interests are in data mining, machine learning and information retrieval. Several of his papers have won best-paper awards, including papers at SIGKDD and SIGMOD. Dr. Keogh is the recipient of a 5-year NSF Career Award for *Efficient Discovery of Previously Unknown Patterns and Relationships in Massive Time Series Databases*.



Chotirat Ann Ratanamahatana is a Ph.D. candidate in computer science at the University of California, Riverside. She received her undergraduate and graduate studies in computer science from Carnegie Mellon University and Harvard University, respectively. Her research interests include data mining, time series classification, information retrieval and human–computer interaction. She is a member of Phi Beta Kappa Honor Society and has been awarded a 10-year full scholarship from the Royal Thai Government since 1993.

Correspondence and offprint requests to: Eamonn Keogh, University of California–Riverside, Computer Science & Engineering Department, Riverside, CA 92521, USA. Email: eamonn@cs.ucr.edu