Faster Retrieval with a Two-Pass Dynamic-Time-Warping Lower Bound

Daniel Lemire [1]

LICEF, Université du Québec à Montréal (UQAM), 100 Sherbrooke West, Montreal (Quebec), H2X 3P2 Canada

Abstract

The Dynamic Time Warping (DTW) is a popular similarity measure between time series. The DTW fails to satisfy the triangle inequality and its computation requires quadratic time. Hence, to find closest neighbors quickly, we use bounding techniques. We can avoid most DTW computations with an inexpensive lower bound (LB_Keogh). We compare LB_Keogh with a tighter lower bound (LB_Improved). We find that LB_Improved-based search is faster. As an example, our approach is 2–3 times faster over random-walk and shape time series.

Key words: time series, very large databases, indexing, classification

1 Introduction

Dynamic Time Warping (DTW) was initially introduced to recognize spoken words [1], but it has since been applied to a wide range of information retrieval and database problems: handwriting recognition [2,3], signature recognition [4,5], image de-interlacing [6], appearance matching for security purposes [7], whale vocalization classification [8], query by humming [9,10], classification of motor activities [11], face localization [12], chromosome classification [13], shape retrieval [14,15], and so on. Unlike the Euclidean distance, DTW optimally aligns or "warps" the data points of two time series (see Fig. 1).

When the distance between two time series forms a metric, such as the Euclidean distance or the Hamming distance, several indexing or search tech-

Email address: `lemire@acm.org` (Daniel Lemire).

[1] phone: 00+1+514 987-3000 ext. 2835, fax: 00+1+514 843-2160

niques have been proposed [16–20]. However, even assuming that we have a metric, Weber et al. have shown that the performance of any indexing scheme degrades to that of a sequential scan, when there are more than a few dimensions [21]. Otherwise—when the distance is not a metric or that the number of dimensions is too large—we use bounding techniques such as the Generic multimedia object indexing (GEMINI) [22]. We quickly discard (most) false positives by computing a lower bound.

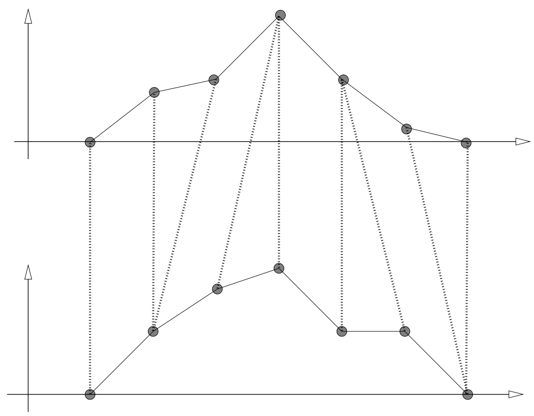


Fig. 1. Dynamic Time Warping example

Ratanamahatana and Keogh [23] argue that their lower bound (LB_Keogh) cannot be improved upon. To make their point, they report that LB_Keogh allows them to prune out over 90% of all DTW computations on several data sets.

We are able to improve upon LB_Keogh as follows. The first step of our two-pass approach is LB_Keogh itself. If this first lower bound is sufficient to discard the candidate, then the computation terminates and the next candidate is considered. Otherwise, we process the time series a second time to increase the lower bound (see Fig. 5). If this second lower bound is large enough, the candidate is pruned, otherwise we compute the full DTW. We show experimentally that the two-pass approach can be several times faster.

The paper is organized as follows. In Section 4, we define the DTW in a generic manner as the minimization of the l_p norm (DTW_p). Among other things, we show that if x and y are separated by a constant ($x \geq c \geq y$ or $x \leq c \leq y$) then the DTW_1 is the l_1 norm (see Proposition 1). In Section 5, we compute generic lower bounds on the DTW and their approximation errors using warping envelopes. In Section 6, we show how to compute the warping envelopes quickly. The next two sections introduce LB_Keogh and LB_Improved respectively. Section 9 presents the application of these lower bounds for multidimensional indexing whereas the last section presents an experimental comparison.

2 Conventions

Time series are arrays of values measured at certain times. For simplicity, we assume a regular sampling rate so that time series are generic arrays of floating-point values. Time series have length n and are indexed from 1 to n. The l_p norm of x is $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$ for any integer $0 < p < \infty$ and $\|x\|_\infty = \max_i |x_i|$. The l_p distance between x and y is $\|x - y\|_p$ and it satisfies the triangle inequality $\|x - z\|_p \leq \|x - y\|_p + \|y - z\|_p$ for $1 \leq p \leq \infty$. The distance between a point x and a set or region S is $d(x, S) = \min_{y \in S} d(x, y)$. Other conventions are summarized in Table 1.

Table 1
Frequently used conventions

n	length of a time series
$\|x\|_p$	l_p norm
DTW_p	monotonic DTW
NDTW_p	non-monotonic DTW
w	DTW locality constraint
$U(x), L(x)$	warping envelope (see Section 5)
$H(x, y)$	projection of x on y (see Equation 1)

3 Related Works

Beside DTW, several similarity metrics have been proposed including the directed and general Hausdorff distance, Pearson's correlation, nonlinear elastic matching distance [24], Edit distance with Real Penalty (ERP) [25], Needleman-Wunsch similarity [26], Smith-Waterman similarity [27], and SimilB [28].

Boundary-based lower-bound functions sometimes outperform LB_Keogh [29]. We can also quantize [30] the time series.

Sakurai et al. [31] have shown that retrieval under the DTW can be faster by mixing progressively finer resolution and by applying early abandoning [32] to the dynamic programming computation.

4 Dynamic Time Warping

A many-to-many matching between the data points in time series x and the data point in time series y matches every data point x_i in x with at least one data point y_j in y, and every data point in y with at least a data point in x. The set of matches (i, j) forms a *warping path* Γ. We define the DTW as the minimization of the l_p norm of the differences $\{x_i - y_j\}_{(i,j)\in\Gamma}$ over all warping paths. A warping path is minimal if there is no subset Γ' of Γ forming an warping path: for simplicity we require all warping paths to be minimal.

In computing the DTW distance, we commonly require the warping to remain local. For time series x and y, we align values x_i and y_j only if $|i - j| \leq w$ for some locality constraint $w \geq 0$ [1]. When $w = 0$, the DTW becomes the l_p distance whereas when $w \geq n$, the DTW has no locality constraint. The value of the DTW diminishes monotonically as w increases. (We do not consider other forms of locality constraints such as the Itakura parallelogram [33].)

Other than locality, DTW can be monotonic: if we align value x_i with value y_j, then we cannot align value x_{i+1} with a value appearing before y_j ($y_{j'}$ for $j' < j$).

We note the DTW distance between x and y using the l_p norm as $\mathrm{DTW}_p(x, y)$ when it is monotonic and as $\mathrm{NDTW}_p(x, y)$ when monotonicity is not required.

By dynamic programming, the monotonic DTW requires $O(wn)$ time. A typical value of w is $n/10$ [23] so that the DTW is in $O(n^2)$. To compute the DTW, we use the following recursive formula. Given an array x, we write the suffix starting at position i, $x_{(i)} = x_i, x_{i+1}, \ldots, x_n$. The symbol $\oplus$ is the exclusive or. Write $q_{i,j} = \mathrm{DTW}_p(x_{(i)}, y_{(j)})^p$ so that $\mathrm{DTW}_p(x, y) = \sqrt[p]{q_{1,1}}$, then

$$q_{i,j} = \begin{cases} 0 & \text{if } |x_{(i)}| = |(y_{(j)}| = 0 \\ \infty & \begin{array}{l}\text{if } |x_{(i)}| = 0 \oplus |y_{(j)}| = 0 \\ \quad\text{or } |i - j| > w\end{array} \\ \begin{array}{c}|x_i - y_j|^p + \\ \min(q_{i+1,j}, q_{i,j+1}, q_{i+1,j+1})\end{array} & \text{otherwise.} \end{cases}$$

For $p = \infty$, we rewrite the preceding recursive formula with $q_{i,j} = \mathrm{DTW}_\infty(x_{(i)}, y_{(j)})$, and $q_{i,j} = \max(|x_i - y_j|, \min(q_{i+1,j}, q_{i,j+1}, q_{i+1,j+1}))$ when $|x_{(i)}| \neq 0$, $|y_{(j)}| \neq 0$, and $|i - j| \leq w$.

We can compute NDTW_1 without locality constraint in $O(n \log n)$ [34]: if the values of the time series are already sorted, the computation is in $O(n)$ time.

We can express the solution of the DTW problem as an alignment of the two initial time series (such as $x = 0,1,1,0$ and $y = 0,1,0,0$) where some of the values are repeated (such as $x' = 0,1,1,0,\mathbf{0}$ and $y' = 0,1,\mathbf{1},0,0$). If we allow non-monotonicity (NDTW), then values can also be inverted.

The non-monotonic DTW is no larger than the monotonic DTW which is itself no larger than the l_p norm: $\mathrm{NDTW}_p(x,y) \le \mathrm{DTW}_p(x,y) \le \|x-y\|_p$ for all $0 < p \le \infty$.

The DTW_1 has the property that if the time series are value-separated, then the DTW is the l_1 norm as the next proposition shows. In Figs. 3 and 4, we present value-separated functions: their DTW_1 is the area between the curves.

Proposition 1 *If x and y are such that either $x \ge c \ge y$ or $x \le c \le y$ for some constant c, then $DTW_1(x,y) = NDTW_1(x,y) = \|x-y\|_1$.*

PROOF. Assume $x \ge c \ge y$. Consider the two aligned (and extended) time series x', y' such that $\mathrm{NDTW}_1(x,y) = \|x'-y'\|_1$. We have that $x' \ge c \ge y'$ and $\mathrm{NDTW}_1(x,y) = \|x'-y'\|_1 = \sum_i |x'_i - y'_i| = \sum_i |x'_i - c| + |c - y'_i| = \|x'-c\|_1 + \|c-y'\|_1 \ge \|x-c\|_1 + \|c-y\|_1 = \|x-y\|_1$. Since we also have $\mathrm{NDTW}_1(x,y) \le \mathrm{DTW}_1(x,y) \le \|x-y\|_1$, the equality follows.

Proposition 1 does not hold for $p > 1$: $\mathrm{DTW}_2((0,0,1,0),(2,3,2,2)) = \sqrt{17}$ whereas $\|(0,0,1,0)-(2,3,2,2)\|_2 = \sqrt{18}$.

5 Computing Lower Bounds on the DTW

Given a time series x, define $U(x)_i = \max_k\{x_k \,|\, |k-i| \le w\}$ and $L(x)_i = \min_k\{x_k \,|\, |k-i| \le w\}$ for $i = 1,\ldots,n$. The pair $U(x)$ and $L(x)$ forms the warping envelope of x (see Fig. 2). We leave the locality constraint w implicit.

The theorem of this section has an elementary proof requiring only the following technical lemma.

Lemma 1 *If $b \in [a,c]$ then $(c-a)^p \ge (c-b)^p + (b-a)^p$ for $1 \le p < \infty$.*

PROOF. For $p = 1$, $(c-b)^p + (b-a)^p = (c-a)^p$. For $p > 1$, by deriving $(c-b)^p + (b-a)^p$ with respect to b, we can show that it is minimized when $b = (c+a)/2$ and maximized when $b \in \{a,c\}$. The maximal value is $(c-a)^p$. Hence the result.

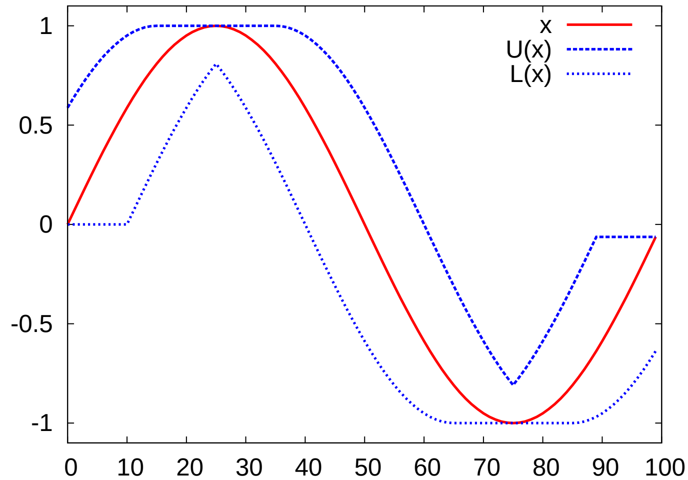


Fig. 2. Warping envelope example

The following theorem introduces a generic result that we use to derive two lower bounds for the DTW including the original Keogh-Ratanamahatana result [35]. Indeed, this new result not only implies the lower bound LB_Keogh, but it also provides a lower bound to the error made by LB_Keogh, thus allowing a tighter lower bound (LB_Improved).

Theorem 1 *Given two equal-length time series x and y and $1 \leq p < \infty$, then for any time series h satisfying $x_i \geq h_i \geq U(y)_i$ or $x_i \leq h_i \leq L(y)_i$ or $h_i = x_i$ for all indexes i, we have*

$$\begin{aligned} DTW_p(x,y)^p &\geq NDTW_p(x,y)^p \\ &\geq \|x-h\|_p^p + NDTW_p(h,y)^p. \end{aligned}$$

For $p = \infty$, a similar result is true: $DTW_\infty(x,y) \geq NDTW_\infty(x,y) \geq \max(\|x - h\|_\infty, NDTW_\infty(h,y))$.

PROOF. Suppose that $1 \leq p < \infty$. Let Γ be a warping path such that $\mathrm{NDTW}_p(x,y)^p = \sum_{(i,j)\in\Gamma} |x_i - y_j|_p^p$. By the constraint on h and Lemma 1, we have that $|x_i - y_j|^p \geq |x_i - h_i|^p + |h_i - y_j|^p$ for any $(i,j) \in \Gamma$ since $h_i \in [\min(x_i, y_j), \max(x_i, y_j)]$. Hence, we have that $\mathrm{NDTW}_p(x,y)^p \geq \sum_{(i,j)\in\Gamma} |x_i - h_i|^p + |h_i - y_j|^p \geq \|x-h\|_p^p + \sum_{(i,j)\in\Gamma} |h_i - y_j|^p$. This proves the result since $\sum_{(i,j)\in\Gamma} |h_i - y_j| \geq \mathrm{NDTW}_p(h,y)$. For $p = \infty$, we have that

$$\begin{aligned} \mathrm{NDTW}_\infty(x,y) &= \max_{(i,j)\in\Gamma} |x_i - y_j| \\ &\leq \max_{(i,j)\in\Gamma} \max(|x_i - h_i|, |h_i - y_j|) \\ &= \max(\|x-h\|_\infty, \mathrm{NDTW}_\infty(h,y)), \end{aligned}$$

concluding the proof.

While Theorem 1 defines a lower bound ($\|x-h\|_p$), the next proposition shows that this lower bound must be a tight approximation as long as h is close to y in the l_p norm.

Proposition 2 *Given two equal-length time series x and y, and $1 \leq p \leq \infty$ with h as in Theorem 1, we have that $\|x-h\|_p$ approximates both $DTW_p(x,y)$ and $NDTW_p(x,y)$ within $\|h-y\|_p$.*

PROOF. By the triangle inequality over l_p, we have $\|x-h\|_p+\|h-y\|_p \geq \|x-y\|_p$. Since $\|x-y\|_p \geq \mathrm{DTW}_p(x,y)$, we have $\|x-h\|_p+\|h-y\|_p \geq \mathrm{DTW}_p(x,y)$, and hence $\|h-y\|_p \geq \mathrm{DTW}_p(x,y) - \|x-h\|_p$. This proves the result since by Theorem 1, we have that $\mathrm{DTW}_p(x,y) \geq \mathrm{NDTW}_p(x,y) \geq \|x-h\|_p$.

This bound on the approximation error is reasonably tight. If x and y are separated by a constant, then $\mathrm{DTW}_1(x,y) = \|x-y\|_1$ by Proposition 1 and $\|x-y\|_1 = \sum_i |x_i - y_i| = \sum_i |x_i - h_i| + |h_i - y_i| = \|x-h\|_1 + \|h-y\|_1$. Hence, the approximation error is exactly $\|h-y\|_1$ in such instances.

6 Warping Envelopes

The computation of the warping envelope $U(x), L(x)$ requires $O(nw)$ time using the naive approach of repeatedly computing the maximum and the minimum over windows. Instead, we compute the envelope with at most $3n$ comparisons between data-point values [36] using Algorithm 1.

7 LB_Keogh

Let $H(x,y)$ be the *projection of x on y* defined as

$$H(x,y)_i = \begin{cases} U(y)_i & \text{if } x_i \geq U(y)_i \\ L(y)_i & \text{if } x_i \leq L(y)_i \\ x_i & \text{otherwise,} \end{cases} \tag{1}$$

for $i = 1, 2, \ldots, n$. We have that $H(x,y)$ is in the envelope of y. By Theorem 1 and setting $h = H(x,y)$, we have that $\mathrm{NDTW}_p(x,y)^p \geq \|x - H(x,y)\|_p^p + \mathrm{NDTW}_p(H(x,y),y)^p$ for $1 \leq p < \infty$. Write $\mathrm{LB\_Keogh}_p(x,y) = \|x - H(x,y)\|_p$ (see Fig. 3), then $\mathrm{LB\_Keogh}_p(x,y)$ is a lower bound to $\mathrm{NDTW}_p(x,y)$ and thus $\mathrm{DTW}_p(x,y)$. The following corollary follows from Theorem 1 and Proposition 2.

Algorithm 1 Streaming algorithm to compute the warping envelope using no more than $3n$ comparisons

input a time series y indexed from 1 to n
input some DTW locality constraint w
return warping envelope U, L (two time series of length n)
u, $l \leftarrow$ empty double-ended queues, we append to "back"
append 1 to u and l
for i in $\{2, \ldots, n\}$ **do**
  if $i \geq w + 1$ **then**
    $U_{i-w} \leftarrow y_{\text{front}(u)}$, $L_{i-w} \leftarrow y_{\text{front}(l)}$
  if $y_i > y_{i-1}$ **then**
    pop u from back
    while $y_i > y_{\text{back}(u)}$ **do**
      pop u from back
  else
    pop l from back
    while $y_i < y_{\text{back}(l)}$ **do**
      pop l from back
  append i to u and l
  if $i = 2w + 1 + \text{front}(u)$ **then**
    pop u from front
  else if $i = 2w + 1 + \text{front}(l)$ **then**
    pop l from front
for i in $\{n + 1, \ldots, n + w\}$ **do**
  $U_{i-w} \leftarrow y_{\text{front}(u)}$, $L_{i-w} \leftarrow y_{\text{front}(l)}$
  if i-front$(u) \geq 2w + 1$ **then**
    pop u from front
  if i-front$(l) \geq 2w + 1$ **then**
    pop l from front

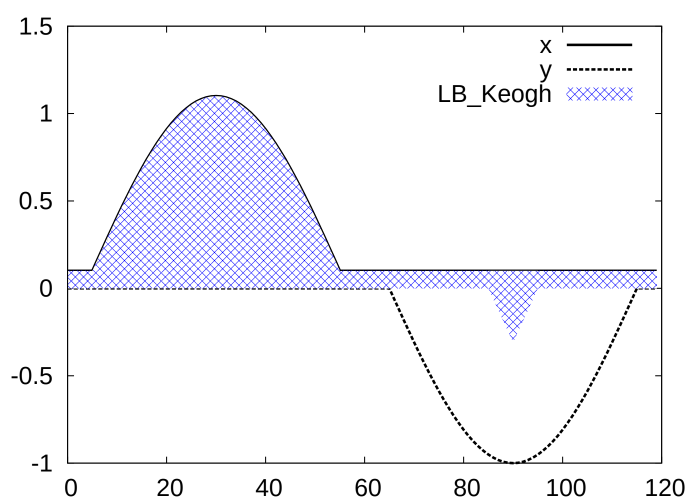


Fig. 3. LB_Keogh example: the area of the marked region is $\text{LB\_Keogh}_1(x, y)$

Corollary 1 *Given two equal-length time series x and y and $1 \leq p \leq \infty$, then*

- *LB_Keogh$_p(x, y)$ is a lower bound to the DTW:*

$$DTW_p(x, y) \geq NDTW_p(x, y) \geq LB\_Keogh_p(x, y);$$

- *the accuracy of LB_Keogh is bounded by the distance to the envelope:*

$$DTW_p(x, y) - LB\_Keogh_p(x, y) \leq \| \max\{U(y)_i - y_i, y_i - L(y)_i\}_i \|_p$$

  for all x.

Algorithm 2 shows how LB_Keogh can be used to find a nearest neighbor in a time series database. We used DTW_1 for all implementations (see Appendix C). The computation of the envelope of the query time series is done once (see line 4). The lower bound is computed in lines 7 to 12. If the lower bound is sufficiently large, the DTW is not computed (see line 13). Ignoring the computation of the full DTW, at most $(2N + 3)n$ comparisons between data points are required to process a database containing N time series.

Algorithm 2 LB_Keogh-based Nearest-Neighbor algorithm

1: **input** a time series y indexed from 1 to n
2: **input** a set S of candidate time series
3: **return** the nearest neighbor B to y in S under DTW_1
4: $U, L \leftarrow \text{envelope}(y)$
5: $b \leftarrow \infty$ {b stores $\min_{x \in S} \mathrm{DTW}_1(x, y)$}
6: **for** candidate x in S **do**
7:     $\beta \leftarrow 0$ {β stores the lower bound}
8:     **for** $i \in \{1, 2, \ldots, n\}$ **do**
9:         **if** $x_i > U_i$ **then**
10:             $\beta \leftarrow \beta + x_i - U_i$
11:         **else if** $x_i < L_i$ **then**
12:             $\beta \leftarrow \beta + L_i - x_i$
13:     **if** $\beta < b$ **then**
14:         $t \leftarrow \mathrm{DTW}_1(a, c)$ {We compute the full DTW.}
15:         **if** $t < b$ **then**
16:             $b \leftarrow t$
17:             $B \leftarrow c$

8 LB_Improved

In the previous Section, we saw that $\mathrm{NDTW}_p(x, y)^p \geq \mathrm{LB\_Keogh}_p(x, y)^p + \mathrm{NDTW}_p(H(x, y), y)^p$ for $1 \leq p < \infty$. In turn, we have $\mathrm{NDTW}_p(H(x, y), y) \geq$

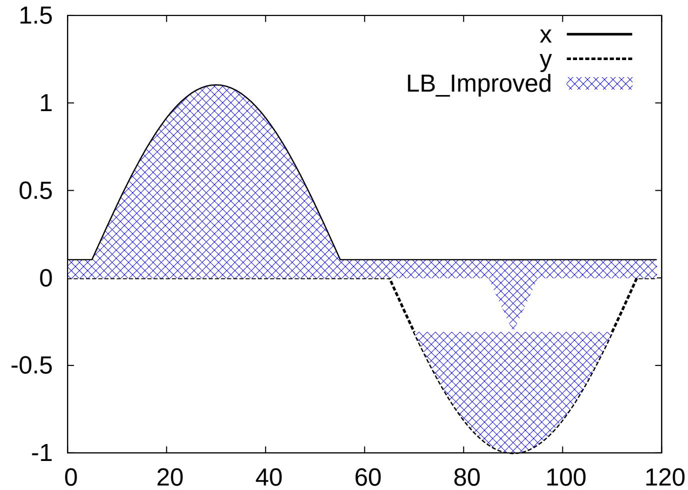


Fig. 4. LB_Improved example: the area of the marked region is $\text{LB\_Improved}_1(x, y)$

$\text{LB\_Keogh}_p(y, H(x, y))$. Hence, write

$$\text{LB\_Improved}_p(x, y)^p = \text{LB\_Keogh}_p(x, y)^p + \text{LB\_Keogh}_p(y, H(x, y))^p$$

for $1 \leq p < \infty$. By definition, we have $\text{LB\_Improved}_p(x, y) \geq \text{LB\_Keogh}_p(x, y)$. Intuitively, whereas $\text{LB\_Keogh}_p(x, y)$ measures the distance between x and the envelope of y, $\text{LB\_Keogh}_p(y, H(x, y))$ measures the distance between y and the envelope of the projection of x on y (see Fig. 4). The next corollary shows that LB_Improved is a lower bound to the DTW.

Corollary 2 *Given two equal-length time series x and y and $1 \leq p < \infty$, then $\text{LB\_Improved}_p(x, y)$ is a lower bound to the DTW: $DTW_p(x, y) \geq NDTW_p(x, y) \geq \text{LB\_Improved}_p(x, y)$.*

PROOF. Recall that $\text{LB\_Keogh}_p(x, y) = \|x - H(x, y)\|_p$. First apply Theorem 1: $\text{DTW}_p(x, y)^p \geq \text{NDTW}_p(x, y)^p \geq \text{LB\_Keogh}_p(x, y)^p + \text{NDTW}_p(H(x, y), y)^p$. Apply Theorem 1 once more: $\text{NDTW}_p(y, H(x, y))^p \geq \text{LB\_Keogh}_p(y, H(x, y))^p$. By substitution, we get $\text{DTW}_p(x, y)^p \geq \text{NDTW}_p(x, y)^p \geq \text{LB\_Keogh}_p(x, y)^p + \text{LB\_Keogh}_p(y, H(x, y))^p$ thus proving the result.

Algorithm 3 shows how to apply LB_Improved as a two-step process (see Fig. 5). Initially, for each candidate x, we compute the lower bound $\text{LB\_Keogh}_1(x, y)$ (see lines 8 to 15). If this lower bound is sufficiently large, the candidate is discarded (see line 16), otherwise we add $\text{LB\_Keogh}_1(y, H(x, y))$ to $\text{LB\_Keogh}_1(x, y)$, in effect computing $\text{LB\_Improved}_1(x, y)$ (see lines 17 to 22). If this larger lower bound is sufficiently large, the candidate is finally discarded (see line 23). Otherwise, we compute the full DTW. If α is the fraction of candidates pruned by

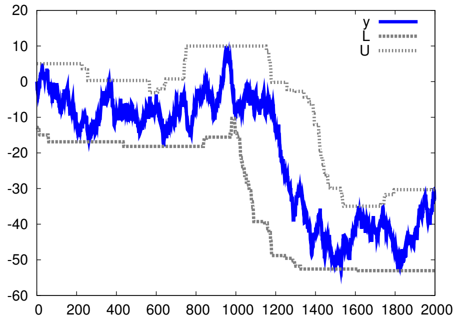
LB_Keogh, at most $(2N+3)n+5(1-\alpha)Nn$ comparisons between data points are required to process a database containing N time series.

Algorithm 3 LB_Improved-based Nearest-Neighbor algorithm

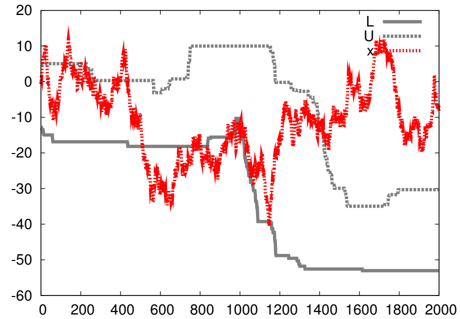
input a time series y indexed from 1 to n
input a set S of candidate time series
return the nearest neighbor B to y in S under DTW_1
$U, L \leftarrow \mathrm{envelope}(y)$
$b \leftarrow \infty$ {b stores $\min_{x\in S} \mathrm{DTW}_1(x, y)$}
for candidate x in S **do**
    copy x to x' {x' will store the projection of x on y}
    $\beta \leftarrow 0$ {β stores the lower bound}
    for $i \in \{1, 2, \ldots, n\}$ **do**
        if $x_i > U_i$ **then**
            $\beta \leftarrow \beta + x_i - U_i$
            $x'_i = U_i$
        else if $x_i < L_i$ **then**
            $\beta \leftarrow \beta + L_i - x_i$
            $x'_i = L_i$
    if $\beta < b$ **then**
        $U', L' \leftarrow \mathrm{envelope}(x')$
        for $i \in \{1, 2, \ldots, n\}$ **do**
            if $y_i > U'_i$ **then**
                $\beta \leftarrow \beta + y_i - U'_i$
            else if $y_i < L'_i$ **then**
                $\beta \leftarrow \beta + L'_i - y_i$
        if $\beta < b$ **then**
            $t \leftarrow \mathrm{DTW}_1(a, c)$ {We compute the full DTW.}
            if $t < b$ **then**
                $b \leftarrow t$
                $B \leftarrow c$

9 Using a multidimensional indexing structure

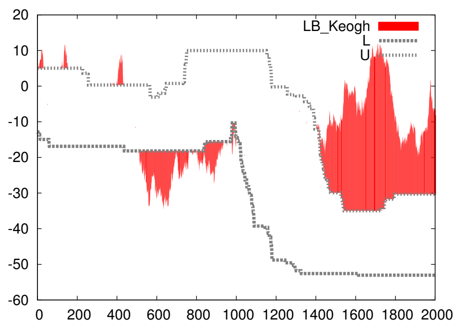
The running time of Algorithms 2 and 3 may be improved if we use a multidimensional index such as an R*-tree [37]. Unfortunately, the performance of such an index diminishes quickly as the number of dimensions increases [21]. To solve this problem, several dimensionality reduction techniques are possible such as piecewise linear [38–40] segmentation. Following Zhu and Shasha [10], we project time series and their envelopes on a d-dimensional space using piecewise sums: $P_d(x) = (\sum_{i\in C_j} x_i)_j$ where $C_1, C_2, \ldots, C_d$ is a disjoint cover of $\{1, 2, \ldots, n\}$. Unlike Zhu and Shasha, we do not require the intervals to have equal length. The l_1 distance between $P_d(y)$ and the minimum bounding hyperrectangle containing $P_d(L(x))$ and $P_d(U(x))$ is a lower bound to the



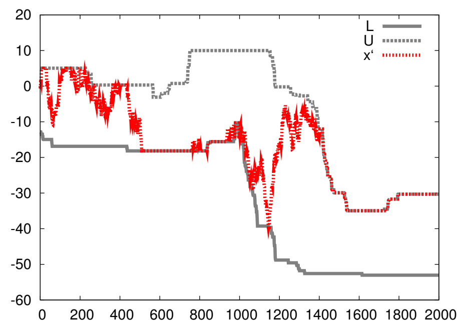
(a) We begin with y and its envelope $L(y), U(y)$.



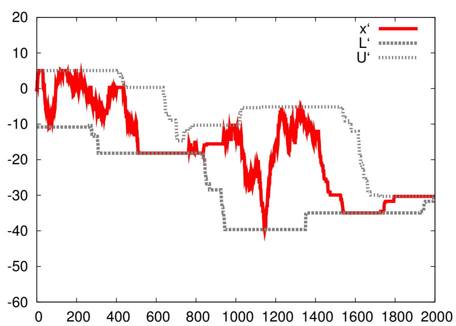
(b) We compare candidate x with the envelope $L(y), U(y)$.



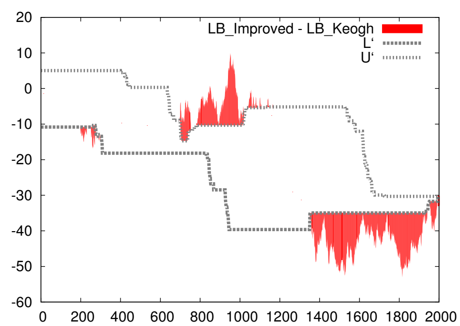
(c) The difference is LB_Keogh(x,y).



(d) We compute x', the projection of x on the envelope $L(y), U(y)$.



(e) We compute the envelope of x'.



(f) The difference between y and the envelope $L(x'), U(x')$ is added to LB_Keogh to compute LB_Improved.

Fig. 5. Computation of LB_Improved as in Algorithm 3

$\mathrm{DTW}_1(x, y)$:

$$\begin{aligned}\mathrm{DTW}_1(x, y) &\geq \mathrm{LB\_Keogh}_1(x, y) \\ &= \sum_{i=1}^{n} d(x_i, [L(y)_i, U(y)_i]) \\ &\geq \sum_{j=1}^{d} d(P_d(x)_j, [P_d(L(y))_j, P_d(U(y))_j]).\end{aligned}$$

For our experiments, we chose the cover $C_j = [1 + (j-1)\lfloor n/d \rfloor, j\lfloor n/d \rfloor]$ for $j = 1, \ldots, d-1$ and $C_d = [1 + (d-1)\lfloor n/d \rfloor, n]$.

We can summarize the Zhu-Shasha R*-tree algorithm as follows:

(1) for each time series x in the database, add $P_d(x)$ to the R*-tree;
(2) given a query time series y, compute its envelope $E = P_d(L(y)), P_d(U(y))$;
(3) starting with $b = \infty$, iterate over all candidate $P_d(x)$ at a l_1 distance b from the envelope E using the R*-tree, once a candidate is found, update b with $\mathrm{DTW}_1(x, y)$ and repeat until you have exhausted all candidates.

This algorithm is correct because the distance between E and $P_d(x)$ is a lower bound to $\mathrm{DTW}_1(x, y)$. However, dimensionality reduction diminishes the pruning power of LB_Keogh : $d(E, P_d(x)) \leq \text{LB\_Keogh}_1(x, y)$. Hence, we propose a new algorithm (R*-Tree+LB_Keogh) where instead of immediately updating b with $\mathrm{DTW}_1(x, y)$, we first compute the LB_Keogh lower bound between x and y. Only when it is less than b, do we compute the full DTW. Finally, as a third algorithm (R*-Tree+LB_Improved), we first compute LB_Keogh, and if it is less than b, then we compute LB_Improved, and only when it is also lower than b do we compute the DTW, as in Algorithm 3. R*-tree+LB_Improved has maximal pruning power, whereas Zhu-Shasha R*-tree has the lesser pruning power of the three alternatives.

10 Comparing Zhu-Shasha R*-tree, LB_Keogh, and LB_Improved

In this section, we benchmark algorithms Zhu-Shasha R*-tree, R*-tree+ LB_Keogh, and R*-tree+LB_Improved. We know that the LB_Improved approach has at least the pruning power of the other methods, but does more pruning translate into a faster nearest-neighbor retrieval under the DTW distance?

We implemented the algorithms in C++ using an external-memory R*-tree. The time series are stored on disk in a binary flat file. We used the GNU GCC 4.0.2 compiler on an Apple Mac Pro, having two Intel Xeon dual-core processors running at 2.66 GHz with 2 GiB of RAM. No thrashing was observed. We measured the wall-clock total time. In all experiments, we benchmark nearest-neighbor retrieval under the DTW_1. By default, the locality constraint w is set at 10% ($w = n/10$). To ensure reproducibility, our source code is freely available [41], including the script used to generate synthetic data sets. We compute the full DTW using a $O(nw)$-time dynamic programming algorithm.

The R*-tree was implemented using the Spatial Index library [42]. In informal tests, we found that a projection on an 8-dimensional space, as described by Zhu and Shasha, gave good results: substantially larger ($d > 10$) or smaller ($d < 6$) settings gave poorer performance. We used a 4,096-byte page size and a 10-entry internal memory buffer.

For R*-TREE+ LB_KEOGH and R*-TREE+LB_IMPROVED, we experimented with early abandoning [32] to cancel the computation of the lower bound as soon as the error is too large. While it often improved retrieval time slightly for both LB_Keogh and LB_Improved, the difference was always small (less than $\approx 1\%$). One explanation is that the candidates produced by the Zhu-Shasha R*-tree are rarely poor enough to warrant efficient early abandoning.

We do not report our benchmarking results over the simple Algorithms 2 and 3. In almost all cases, the R*-tree equivalent—R*-TREE+ LB_KEOGH or R*-TREE+LB_IMPROVED—was at least slightly better and sometimes several times faster.

10.1 Synthetic data sets

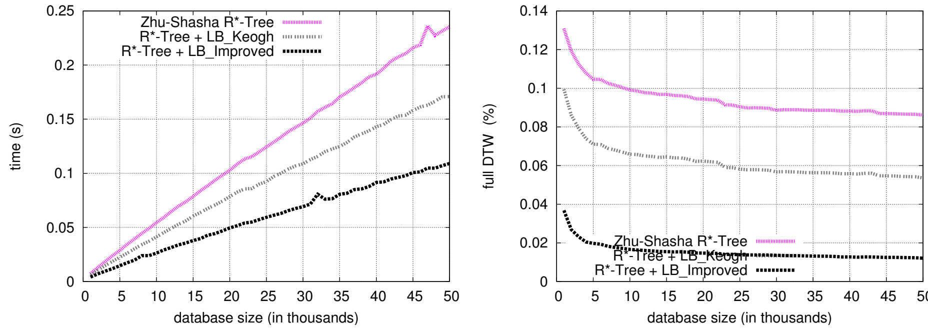
We tested our algorithms using the Cylinder-Bell-Funnel [43] and Control Charts [44] data sets, as well as over two databases of random walks. We generated 256-sample and 1 000-sample random-walk time series using the formula $x_i = x_{i-1} + N(0, 1)$ and $x_1 = 0$.

For each data set, we generated a database of 50 000 time series by adding randomly chosen items. Figs. 6, 7, 8 and 9 show the average timings and pruning ratio averaged over 20 queries based on randomly chosen time series as we consider larger and large fraction of the database. LB_Improved prunes between 2 and 4 times more candidates than LB_Keogh. R*-TREE+LB_IMPROVED is faster than Zhu-Shasha R*-tree by a factor between 0 and 6.

We saw almost no performance gain over Zhu-Shasha R*-tree with simple time series such as the Cylinder-Bell-Funnel or the Control Charts data sets. However, in these cases, even LB_Improved has modest pruning powers of 40% and 15%. Low pruning means that the computational cost is dominated by the cost of the full DTW.

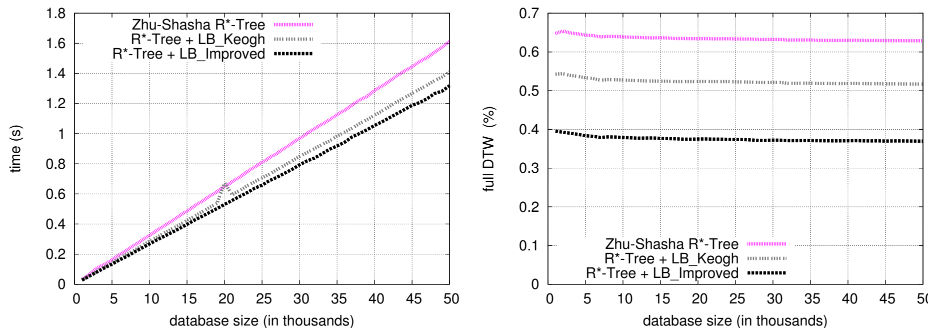
10.2 Shape data sets

We also considered a large collection of time-series derived from shapes [45,46]. The first data set is made of heterogeneous shapes which resulted in 5 844 1 024-sample times series. The second data set is an arrow-head data set



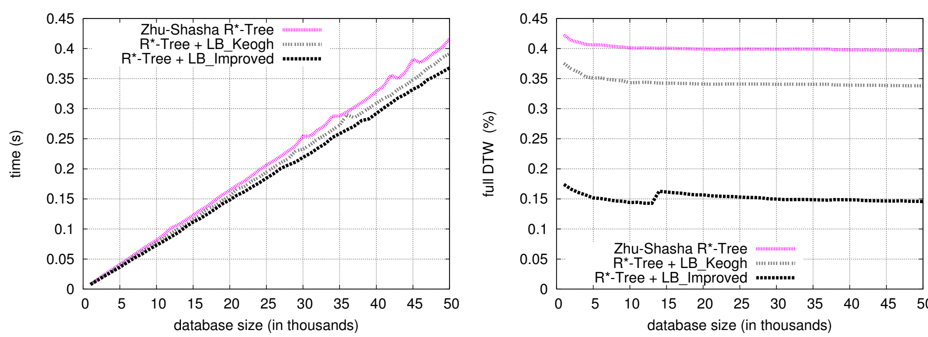
(a) Average Retrieval Time (b) Pruning Power

Fig. 6. Nearest-Neighbor Retrieval for the 256-sample random-walk data set



(a) Average Retrieval Time (b) Pruning Power

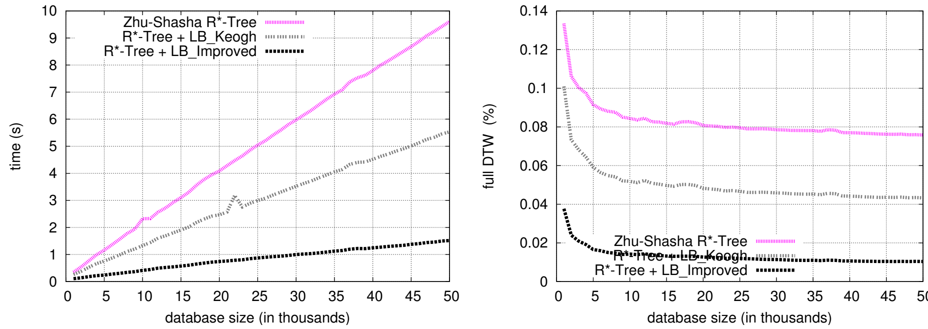
Fig. 7. Nearest-Neighbor Retrieval for the Cylinder-Bell-Funnel data set



(a) Average Retrieval Time (b) Pruning Power

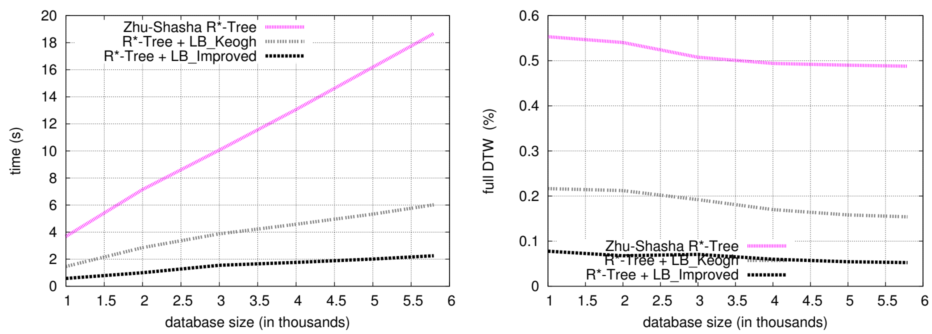
Fig. 8. Nearest-Neighbor Retrieval for the Control Charts data set

with of 15 000 251-sample time series. We extracted 50 time series from each data set, and we present the average nearest-neighbor retrieval times and pruning power as we consider various fractions of each database (see Figs. 10 and 11). The results are similar: LB_Improved has twice the pruning power than LB_Keogh, R*-tree+LB_Improved is twice as fast as R*-tree+LB_Keogh and over 3 times faster than the Zhu-Shasha R*-tree.



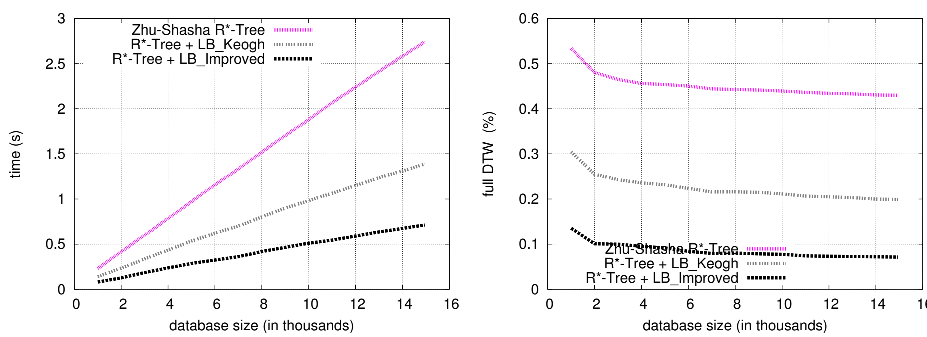
(a) Average Retrieval Time (b) Pruning Power

Fig. 9. Nearest-Neighbor Retrieval for the 1000-sample random-walk data set



(a) Average Retrieval Time (b) Pruning Power

Fig. 10. Nearest-Neighbor Retrieval for the heterogeneous shape data set

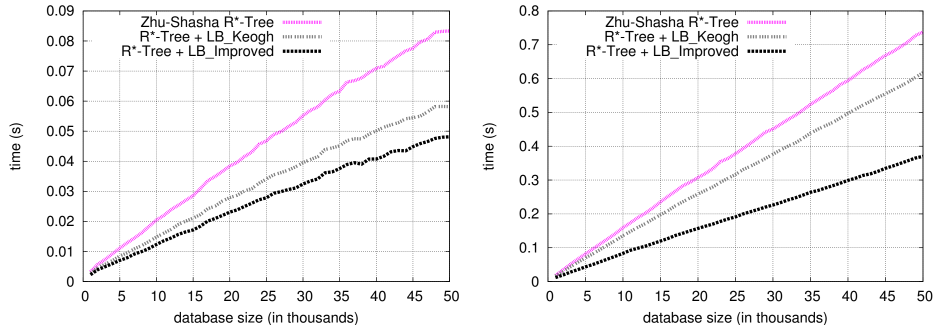


(a) Average Retrieval Time (b) Pruning Power

Fig. 11. Nearest-Neighbor Retrieval for the arrow-head shape data set

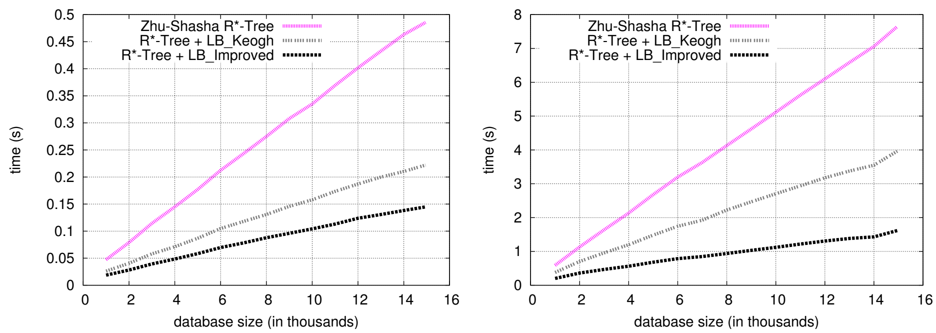
10.3 Locality constraint

The locality constraint has an effect on retrieval times: a large value of w makes the problem more difficult and reduces the pruning power of all methods. In Figs. 12 and 13, we present the retrieval times for $w = 5\%$ and $w = 20\%$. The benefits of R*-TREE+LB_IMPROVED remain though they are less significant for small locality constraints. Nevertheless, even in this case, R*-



(a) $w = 5\%$ (b) $w = 20\%$

Fig. 12. Average Nearest-Neighbor Retrieval Time for the 256-sample random-walk data set



(a) $w = 5\%$ (b) $w = 20\%$

Fig. 13. Average Nearest-Neighbor Retrieval Time for the arrow-head shape data set

TREE+LB_IMPROVED can still be three times faster than Zhu-Shasha R*-tree. For all our data sets and for all values of $w \in \{5\%, 10\%, 20\%\}$, R*-TREE+LB_IMPROVED was always at least as fast as the Zhu-Shasha R*-tree algorithm alone.

11 Conclusion

We have shown that a two-pass pruning technique can improve the retrieval speed by three times or more in several time-series databases. In our implementation, LB_Improved required slightly more computation than LB_Keogh, but its added pruning power was enough to make the overall computation several times faster. Moreover, we showed that pruning candidates left from the Zhu-Shasha R*-tree with the full LB_Keogh alone—without dimensionality reduction—was enough to significantly boost the speed and pruning power. On some synthetic data sets, neither LB_Keogh nor LB_Improved were able to prune enough candidates, making all algorithms comparable in speed.

Acknowledgements

The author is supported by NSERC grant 261437 and FQRNT grant 112381.

References

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

[2] C. Bahlmann, The writer independent online handwriting recognition system frog on hand and cluster generative statistical Dynamic Time Warping, Writer 26 (3) (2004) 299–310.

[3] R. Niels, L. Vuurpijl, Using Dynamic Time Warping for intuitive handwriting recognition, in: IGS2005, 2005, pp. 217–221.

[4] M. Faundez-Zanuy, On-line signature recognition based on VQ-DTW, Pattern Recogn. 40 (3) (2007) 981–992.

[5] W. Chang, J. Shin, Modified Dynamic Time Warping for stroke-based on-line signature verification, in: ICDAR 2007, 2007, pp. 724–728.

[6] A. Almog, A. Levi, A. M. Bruckstein, Spatial de-interlacing using dynamic time warping, in: ICIP 2005, Vol. 2, 2005, pp. 1010–1013.

[7] A. Kale, N. Cuntoor, B. Yegnanarayana, A. N. Rajagopalan, R. Chellappa, Optical and Digital Techniques for Information Security, Springer-Verlag, 2004, Ch. Gait-Based Human Identification Using Appearance Matching, pp. 271–295.

[8] J. C. Brown, A. Hodgins-Davis, P. J. O. Miller, Classification of vocalizations of killer whales using dynamic time warping, J. Acoust. Soc. Am 119 (3) (2006) 34–40.

[9] J.-S. R. Jang, H.-R. Lee, A general framework of progressive filtering and its application to query by singing/humming, IEEE Transactions on Audio, Speech, and Language Processing 16 (2) (2008) 350–358.

[10] Y. Zhu, D. Shasha, Warping indexes with envelope transforms for query by humming, in: SIGMOD'03, 2003, pp. 181–192.

[11] R. Muscillo, S. Conforto, M. Schmid, P. Caselli, T. D'Alessio, Classification of motor activities through derivative dynamic time warping applied on accelerometer data, in: EMBS 2007, 2007, pp. 4930–4933.

[12] L. E. M. Lopez, R. P. Elias, J. V. Tavira, Face localization in color images using dynamic time warping and integral projections, in: IJCNN 2007, 2007, pp. 892–896.

[13] B. Legrand, C. S. Chang, S. H. Ong, S. Y. Neo, N. Palanisamy, Chromosome classification using dynamic time warping, Pattern Recognition Letters 29 (3) (2007) 215–222.

[14] I. Bartolini, P. Ciaccia, M. Patella, WARP: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (1) (2005) 142–147.

[15] A. Marzal, V. Palazon, G. Peris, Contour-based shape retrieval using Dynamic Time Warping, Lecture notes in Computer Science 4177 (2006) 190.

[16] E. Chávez, G. Navarro, R. Baeza-Yates, J. L. Marroquín, Searching in metric spaces, ACM Comput. Surv. 33 (3) (2001) 273–321.

[17] K. Fredriksson, Engineering efficient metric indexes, Pattern Recogn. Lett. 28 (1) (2007) 75–84.

[18] G. R. Hjaltason, H. Samet, Index-driven similarity search in metric spaces (survey article), ACM Trans. Database Syst. 28 (4) (2003) 517–580.

[19] L. Micó, J. Oncina, R. C. Carrasco, A fast branch & bound nearest neighbour classifier in metric spaces, Pattern Recogn. Lett. 17 (7) (1996) 731–739.

[20] J. Z. C. Lai, Y.-C. Liaw, J. Liu, Fast k-nearest-neighbor search based on projection and triangular inequality, Pattern Recogn. 40 (2) (2007) 351–359.

[21] R. Weber, H.-J. Schek, S. Blott, A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, in: VLDB '98, 1998, pp. 194–205.

[22] C. Faloutsos, Searching Multimedia Databases by Content, Kluwer Academic Publishers, 1996.

[23] C. A. Ratanamahatana, E. Keogh, Three myths about Dynamic Time Warping data mining, in: SDM'05, 2005.

[24] R. C. Veltkamp, Shape matching: similarity measures and algorithms, in: Shape Modeling and Applications, 2001, pp. 188–197.

[25] L. Chen, R. Ng, On the marriage of l_p-norms and edit distance, in: VLDB'04, 2004, pp. 1040–1049.

[26] S. B. Needleman, C. D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol. 48 (3) (1970) 443–53.

[27] T. F. Smith, M. S. Waterman, Identification of common molecular subsequences, J. Mol. Bwl 147 (1981) 195–197.

[28] A.-O. Boudraa, J.-C. Cexus, M. Groussat, P. Brunagel, An energy-based similarity measure for time series, EURASIP J. Adv. Signal Process 2008 (1) (2008) 1–9.

[29] M. Zhou, M. H. Wong, Boundary-based lower-bound functions for Dynamic Time Warping and their indexing, ICDE 2007 (2007) 1307–1311.

[30] I. F. Vega-López, B. Moon, Quantizing time series for efficient similarity search under time warping, in: ACST'06, ACTA Press, Anaheim, CA, USA, 2006, pp. 334–339.

[31] Y. Sakurai, M. Yoshikawa, C. Faloutsos, FTW: fast similarity search under the time warping distance, in: PODS '05, 2005, pp. 326–337.

[32] L. Wei, E. Keogh, H. V. Herle, A. Mafra-Neto, Atomic wedgie: Efficient query filtering for streaming times series, in: ICDM '05, 2005, pp. 490–497.

[33] F. Itakura, Minimum prediction residual principle applied to speech recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 23 (1) (1975) 67–72.

[34] J. Colannino, M. Damian, F. Hurtado, S. Langerman, H. Meijer, S. Ramaswami, D. Souvaine, G. Toussaint, Efficient many-to-many point matching in one dimension, Graph. Comb. 23 (1) (2007) 169–178.

[35] E. Keogh, C. A. Ratanamahatana, Exact indexing of dynamic time warping, Knowledge and Information Systems 7 (3) (2005) 358–386.

[36] D. Lemire, Streaming maximum-minimum filter using no more than three comparisons per element, Nordic Journal of Computing 13 (4) (2006) 328–339.

[37] N. Beckmann, H. Kriegel, R. Schneider, B. Seeger, The R*-tree: an efficient and robust access method for points and rectangles, SIGMOD '90 (1990) 322–331.

[38] H. Xiao, X.-F. Feng, Y.-F. Hu, A new segmented time warping distance for data mining in time series database, in: Machine Learning and Cybernetics 2004, Vol. 2, 2004, pp. 1277–1281.

[39] Y. Shou, N. Mamoulis, D. W. Cheung, Fast and exact warping of time series using adaptive segmental approximations, Mach. Learn. 58 (2-3) (2005) 231–267.

[40] X. L. Dong, C. K. Gu, Z. O. Wang, A local segmented Dynamic Time Warping distance measure algorithm for time series data mining, in: International Conference on Machine Learning and Cybernetics 2006, 2006, pp. 1247–1252.

[41] D. Lemire, Fast nearest-neighbor retrieval under the dynamic time warping, online: `http://code.google.com/p/lbimproved/` (2008).

[42] M. Hadjieleftheriou, Spatial index library, online: `http://research.att.com/~marioh/spatialindex/` (2008).

[43] N. Saito, Local feature extraction and its applications using a library of bases, Ph.D. thesis, Yale University, New Haven, CT, USA (1994).

[44] D. T. Pham, A. B. Chan, Control chart pattern recognition using a new type of self-organizing neural network, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 212 (2) (1998) 115–127.

[45] E. Keogh, L. Wei, X. Xi, S. H. Lee, M. Vlachos, LB_Keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures, VLDB 2006 (2006) 882–893.

[46] E. Keogh, Shape matching, online: `http://www.cs.ucr.edu/~eamonn/shape/shape.htm`, papers and data sets (2007).

[47] G. B. Folland, Real Analysis. Modern Techniques and Their Applications, Wiley, 1984.

[48] F. Casacuberta, E. Vidal, H. Rulot, On the metric properties of dynamic time warping, IEEE Transactions on Acoustics, Speech, and Signal Processing 35 (11) (1987) 1631–1633.

[49] T. Eiter, H. Mannila, Computing discrete frechet distance, Tech. Rep. CD-TR 94/64, Christian Doppler Laboratory for Expert Systems (1994).

[50] M. D. Morse, J. M. Patel, An efficient and accurate method for evaluating time series similarity, Proceedings of the 2007 ACM SIGMOD international conference on Management of data (2007) 569–580.

[51] C. C. Aggarwal, A. Hinneburg, D. A. Keim, On the surprising behavior of distance metrics in high dimensional spaces, in: ICDT'01, 2001, pp. 420–434.

[52] D. François, V. Wertz, M. Verleysen, The concentration of fractional distances, IEEE Transactions on Knowledge and Data Engineering 19 (7) (2007) 873–886.

[53] L. Breiman, Classification and Regression Trees, Chapman & Hall/CRC, 1998.

[54] C. A. Gonzalez, J. J. R. Diez, Time series classification by boosting interval based literals, Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial 11 (2000) 2–11.

About the Author–DR. DANIEL LEMIRE received a B.Sc. and a M.Sc. in Mathematics from the University of Toronto in 1994 and 1995. He received his Ph.D. in Engineering Mathematics from the Ecole Polytechnique and the Université de Montréal in 1998. He completed a post-doctoral fellowship at the Institut de génie biomédical and worked as consultant in industry. From 2002 to 2004, he was a research officer at the National Research Council of Canada (NRC). He is now a professor at the Université du Québec à Montréal (UQAM) where he teaches Computer Science. His research interests include data warehousing, OLAP and time series.

A Some Properties of Dynamic Time Warping

The DTW distance can be counterintuitive. As an example, if x, y, z are three time series such that $x \leq y \leq z$ pointwise, then it does not follow that

$\mathrm{DTW}_p(x,z) \geq \mathrm{DTW}_p(z,y)$. Indeed, choose $x = 7,0,1,0$, $y = 7,0,5,0$, and $z = 7,7,7,0$, then $\mathrm{DTW}_\infty(z,y) = 5$ and $\mathrm{DTW}_\infty(z,x) = 1$. Hence, we review some of the mathematical properties of the DTW.

The warping path aligns x_i from time series x and y_j from time series y if $(i,j) \in \Gamma$. The next proposition is a general constraint on warping paths.

Proposition 3 *Consider any two time series x and y. For any minimal warping path, if x_i is aligned with y_j, then either x_i is aligned only with y_j or y_j is aligned only with x_i. Therefore the length of a minimal warping path is at most $2n-2$ when $n > 1$.*

PROOF. Suppose that the result is not true. Then there is x_k, x_i and y_l, y_j such that x_k and x_i are aligned with y_j, and y_l and y_j are aligned with x_i. We can delete (k,j) from the warping path and still have a warping path. A contradiction.

Next, we show that warping path is no longer than $2n-2$. Let n_1 be the number of points in x aligned with only one point in y, and let n_2 be the number of points in y aligned with only one point in x. The cardinality of a minimal warping path is bounded by $n_1 + n_2$. If $n_1 = n$ or $n_2 = n$, then $n_1 = n_2 = n$ and the warping path has cardinality n which is no larger than $2n-2$ for $n > 1$. Otherwise, $n_1 \leq n-1$ and $n_2 \leq n-1$, and $n_1 + n_2 < 2n-2$.

The next lemma shows that the DTW becomes the l_p distance when either x or y is constant.

Lemma 2 *For any $0 < p \leq \infty$, if $y = c$ is a constant, then $NDTW_p(x,y) = DTW_p(x,y) = \|x-y\|_p$.*

When $p = \infty$, a stronger result is true: if $y = x + c$ for some constant c, then $\mathrm{NDTW}_\infty(x,y) = \mathrm{DTW}_\infty(x,y) = \|x-y\|_\infty$. Indeed, $\mathrm{NDTW}_\infty(x,y) \geq |\max(y) - \max(x)| = c = \|x-y\|_\infty \geq \|x-y\|_\infty$ which shows the result. This same result is not true for $p < \infty$: for $x = 0,1,2$ and $y = 1,2,3$, we have $\|x-y\|_p = \sqrt[p]{3}$ whereas $\mathrm{DTW}_p(x,y) = \sqrt[p]{2}$. However, the DTW is translation invariant: $\mathrm{DTW}_p(x,z) = \mathrm{DTW}_p(x+b, z+b)$ and $\mathrm{NDTW}_p(x,z) = \mathrm{NDTW}_p(x+b, z+b)$ for any scalar b and $0 < p \leq \infty$.

In classical analysis, we have that $n^{1/p-1/q}\|x\|_q \geq \|x\|_p$ [47] for $1 \leq p < q \leq \infty$. A similar results is true for the DTW and it allows us to conclude that $\mathrm{DTW}_p(x,y)$ and $\mathrm{NDTW}_p(x,y)$ decrease monotonically as p increases.

Proposition 4 *For $1 \leq p < q \leq \infty$, we have that $(2n-2)^{1/p-1/q} DTW_q(x,y) \geq$*

$DTW_p(x,y)$ where n is the length of x and y. The result also holds for the non-monotonic DTW.

PROOF. Assume $n > 1$. The argument is the same for the monotonic or non-monotonic DTW. Given x, y consider the two aligned (and extended) time series x', y' such that $\mathrm{DTW}_q(x,y) = \|x' - y'\|_q$. Let $n_{x'}$ be the length of x' and $n_{y'}$ be the length of y'. As a consequence of Proposition 3, we have $n_{x'} = n_{y'} \leq 2n - 2$. From classical analysis, we have $n_{x'}^{1/p-1/q}\|x' - y'\|_q \geq \|x' - y'\|_p$, hence $|2n-2|^{1/p-1/q}\|x'-y'\|_q \geq \|x'-y'\|_p$ or $|2n-2|^{1/p-1/q}\mathrm{DTW}_q(x,y) \geq \|x'-y'\|_p$. Since x', y' represent a valid warping path of x, y, then $\|x'-y'\|_p \geq \mathrm{DTW}_p(x,y)$ which concludes the proof.

B The Triangle Inequality

The DTW is commonly used as a similarity measure: x and y are similar if $\mathrm{DTW}_p(x,y)$ is small. Similarity measures often define equivalence relations: $A \sim A$ for all A (reflexivity), $A \sim B \Rightarrow B \sim A$ (symmetry) and $A \sim B \wedge B \sim C \Rightarrow A \sim C$ (transitivity).

The DTW is reflexive and symmetric, but it is not transitive. Indeed, consider the following time series:

$$\begin{aligned}
X &= \underbrace{0, 0, \ldots, 0, 0}_{2m+1 \text{ times}},\\
Y &= \underbrace{0, 0, \ldots, 0, 0}_{m \text{ times}}, \epsilon, \underbrace{0, 0, \ldots, 0, 0}_{m \text{ times}},\\
Z &= 0, \underbrace{\epsilon, \epsilon, \ldots, \epsilon, \epsilon}_{2m-1 \text{ times}}, 0.
\end{aligned}$$

We have that $\mathrm{NDTW}_p(X,Y) = \mathrm{DTW}_p(X,Y) = |\epsilon|$, $\mathrm{NDTW}_p(Y,Z) = \mathrm{DTW}_p(Y,Z) = 0$, $\mathrm{NDTW}_p(X,Z) = \mathrm{DTW}_p(X,Z) = \sqrt[p]{(2m-1)}|\epsilon|$ for $1 \leq p < \infty$ and $w = m - 1$. Hence, for ϵ small and $n \gg 1/\epsilon$, we have that $X \sim Y$ and $Y \sim Z$, but $X \not\sim Z$. This example proves the following lemma.

Lemma 3 *For $1 \leq p < \infty$ and $w > 0$, neither DTW_p nor $NDTW_p$ satisfies a triangle inequality of the form $d(x,y) + d(y,z) \geq cd(x,z)$ where c is independent of the length of the time series and of the locality constraint.*

This theoretical result is somewhat at odd with practical experience. Casacuberta et al. found no triangle inequality violation in about 15 million triplets of voice recordings [48]. To determine whether we could expect violations of the triangle inequality in practice, we ran the following experiment. We used 3 types of 100-sample time series: white-noise times series defined by $x_i = N(0,1)$ where N is the normal distribution, random-walk time series defined by $x_i = x_{i-1} + N(0,1)$ and $x_1 = 0$, and the Cylinder-Bell-Funnel time series proposed by Saito [43]. For each type, we generated 100 000 triples of time series x, y, z and we computed the histogram of the function

$$C(x,y,z) = \frac{\mathrm{DTW}_p(x,z)}{\mathrm{DTW}_p(x,y) + \mathrm{DTW}_p(y,z)}$$

for $p = 1$ and $p = 2$. The DTW is computed without time constraints. Over the white-noise and Cylinder-Bell-Funnel time series, we failed to find a single violation of the triangle inequality: a triple x, y, z for which $C(x,y,z) > 1$. However, for the random-walk time series, we found that 20% and 15% of the triples violated the triangle inequality for DTW_1 and DTW_2.

The DTW satisfies a weak triangle inequality as the next theorem shows.

Theorem 2 *Given any 3 same-length time series x, y, z and $1 \leq p \leq \infty$, we have*

$$DTW_p(x,y) + DTW_p(y,z) \geq \frac{DTW_p(x,z)}{\min(2w+1,n)^{1/p}}$$

where w is the locality constraint. The result also holds for the non-monotonic DTW.

PROOF. Let Γ and Γ' be minimal warping paths between x and y and between y and z. Let $\Gamma'' = \{(i,j,k) | (i,j) \in \Gamma \text{ and } (j,k) \in \Gamma'\}$. Iterate through the tuples (i,j,k) in Γ'' and construct the same-length time series x'', y'', z'' from x_i, y_j, and z_k. By the locality constraint any match $(i,j) \in \Gamma$ corresponds to at most $\min(2w+1,n)$ tuples of the form $(i,j,\cdot) \in \Gamma''$, and similarly for any match $(j,k) \in \Gamma'$. Assume $1 \leq p < \infty$. We have that $\|x''-y''\|_p^p = \sum_{(i,j,k)\in\Gamma''} |x_i - y_j|^p \leq \min(2w+1,n)\mathrm{DTW}_p(x,y)^p$ and $\|y''-z''\|_p^p = \sum_{(i,j,k)\in\Gamma''} |y_j - z_k|^p \leq \min(2w+1,n)\mathrm{DTW}_p(y,z)^p$. By the triangle inequality in l_p, we have

$$\begin{aligned} \min(2w+1,n)^{1/p}(\mathrm{DTW}_p(x,y) + \mathrm{DTW}_p(y,z)) &\geq \|x''-y''\|_p + \|y''-z''\|_p \\ &\geq \|x''-z''\|_p \geq \mathrm{DTW}_p(x,z). \end{aligned}$$

For $p = \infty$, $\max_{(i,j,k)\in\Gamma''} \|x_i - y_j\|_p^p = \mathrm{DTW}_\infty(x,y)^p$ and $\max_{(i,j,k)\in\Gamma''} |y_j - z_k|^p = \mathrm{DTW}_\infty(y,z)^p$, thus proving the result by the triangle inequality over l_∞. The proof is the same for the non-monotonic DTW.

The constant $\min(2w+1,n)^{1/p}$ is tight. Consider the example with time series X, Y, Z presented before Lemma 3. We have $\mathrm{DTW}_p(X,Y) + \mathrm{DTW}_p(Y,Z) = |\epsilon|$

and $\mathrm{DTW}_p(X, Z) = \sqrt[p]{(2w+1)}|\epsilon|$. Therefore, we have

$$\mathrm{DTW}_p(X, Y) + \mathrm{DTW}_p(Y, Z) = \frac{\mathrm{DTW}_p(X, Z)}{\min(2w+1, n)^{1/p}}.$$

A consequence of this theorem is that DTW_∞ satisfies the traditional triangle inequality.

Corollary 3 *The triangle inequality $d(x, y) + d(y, z) \geq d(x, z)$ holds for DTW_∞ and $NDTW_\infty$.*

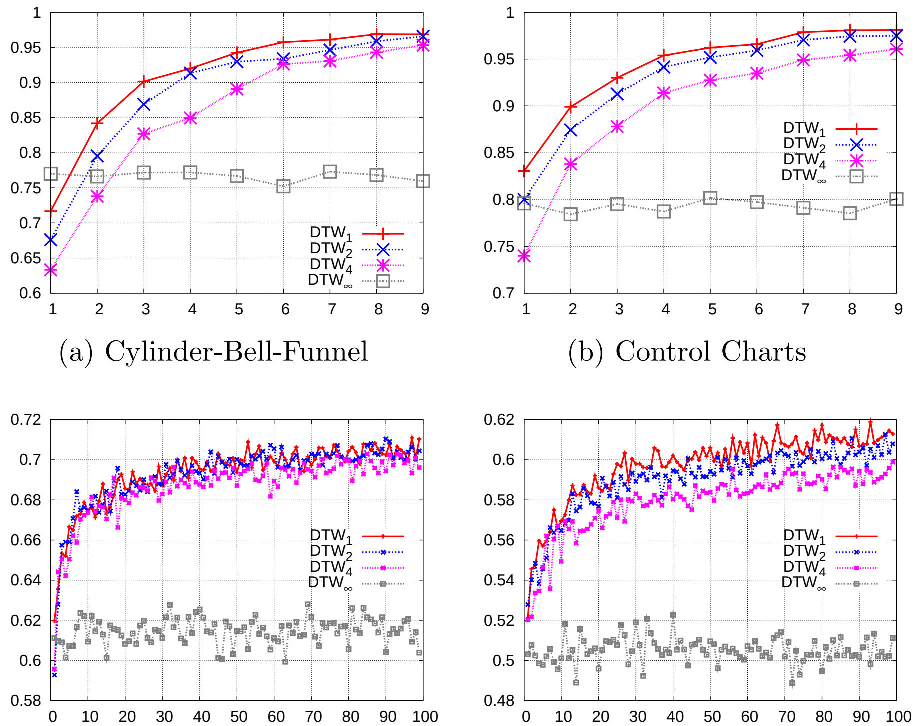
Hence the DTW_∞ is a pseudometric: it is a metric over equivalence classes defined by $x \sim y$ if and only if $\mathrm{DTW}_\infty(x, y) = 0$. When no locality constraint is enforced ($w \geq n$), DTW_∞ is equivalent to the discrete Fréchet distance [49].

C Which is the Best Distance Measure?

The DTW can be seen as the minimization of the l_p distance under warping. Which p should we choose? Legrand et al. reported best results for chromosome classification using DTW_1 [13] as opposed to using DTW_2. However, they did not quantify the benefits of DTW_1. Morse and Patel reported similar results with both DTW_1 and DTW_2 [50].

While they do not consider the DTW, Aggarwal et al. [51] argue that out of the usual l_p norms, only the l_1 norm, and to a lesser extend the l_2 norm, express a qualitatively meaningful distance when there are numerous dimensions. They even report on classification-accuracy experiments where fractional l_p distances such as $l_{0.1}$ and $l_{0.5}$ fare better. François et al. [52] made the theoretical result more precise showing that under uniformity assumptions, lesser values of p are always better.

To compare DTW_1, DTW_2, DTW_4 and DTW_∞, we considered four different synthetic time-series data sets: Cylinder-Bell-Funnel [43], Control Charts [44], Waveform [53], and Wave+Noise [54]. The time series in each data sets have lengths 128, 60, 21, and 40. The Control Charts data set has 6 classes of time series whereas the other 3 data sets have 3 classes each. For each data set, we generated various databases having a different number of instances per class: between 1 and 9 inclusively for Cylinder-Bell-Funnel and Control Charts, and between 1 and 99 for Waveform and Wave+Noise. For a given data set and a given number of instances, 50 different databases were generated. For each database, we generated 500 new instances chosen from a random class and we found a nearest neighbor in the database using DTW_p for $p = 1, 2, 4, \infty$ and



(a) Cylinder-Bell-Funnel

(b) Control Charts

(c) Waveform

(d) Wave+Noise

Fig. C.1. Classification accuracy versus the number of instances of each class in four data sets

using a time constraint of $w = n/10$. When the instance is of the same class as the nearest neighbor, we considered that the classification was a success.

The average classification accuracies for the 4 data sets, and for various number of instances per class is given in Fig. C.1. The average is taken over 25 000 classification tests (50×500), over 50 different databases.

Only when there are one or two instances of each class is DTW_∞ competitive. Otherwise, the accuracy of the DTW_∞-based classification does not improve as we add more instances of each class. For the Waveform data set, DTW_1 and DTW_2 have comparable accuracies. For the other 3 data sets, DTW_1 has a better nearest-neighbor classification accuracy than DTW_2. Classification with DTW_4 has almost always a lower accuracy than either DTW_1 or DTW_2.

Based on these results, DTW_1 is a good choice to classify time series whereas DTW_2 is a close second.