Fast Subsequence Search on Time Series Data

Yuhong Li[1]*, Bo Tang[2]*, Leong Hou U[1], Man Lung Yiu[2], Zhiguo Gong[1]

[1]Department of Computer and Information Science, University of Macau
{yb27407,ryanlhu,fstzgg}@umac.mo

[2]Department of Computing, Hong Kong Polytechnic University
{csbtang,csmlyiu}@comp.polyu.edu.hk

ABSTRACT

Many applications on time series data require solving the subsequence search problem, which has been extensively studied in the database and data mining communities. These applications are computation bound rather than disk I/O bound. In this work, we further propose effective and cheap lower-bounds to reduce the computation cost of the subsequence search problem. Experimental studies show that the proposed lower-bounds can boost the performance of the state-of-the-art solution by up to an order of magnitude.

1. INTRODUCTION

The subsequence search problem on time series data has extensive applications in medical diagnosis, speech processing, climate analysis, financial analysis, etc [1, 5]. Specifically, given a query time series q and a target time series t, the subsequence search problem finds a subsequence t_c of t such that it has the smallest distance $dist(q, t_c)$ to q. Figure 1 illustrates the subsequence search problem. The typical distance measures are the Euclidean distance (ED) and Dynamic Time Warping (DTW).

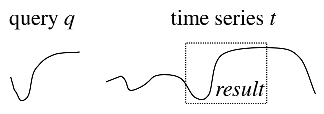


Figure 1: Subsequence search on time series

Euclidean Distance (ED) is the most common similarity measure [1, 5] due to its simplicity and applicability. The distance between two time series of length m is given as follows.

$$ED(q, t_c) = \sqrt{\sum_{i=1}^{m}(q[i] - t_c[i])^2} \quad (1)$$

*indicates equal contribution

Dynamic Time Warping (DTW) is proposed to capture the similarity of two sequences which may vary in time or have missing values. It has been shown to be an effective distance measure [2]. This distance is defined as $DTW(q, t_c) = \sqrt{DTW_{SQ}(q, t_c)}$, where $DTW_{SQ}(q, t_c)$ is computed as follows.

$$DTW_{SQ}(q, t_c) = (q[1] - t_c[1])^2 + \min \begin{cases} DTW_{SQ}(q[2...last], t_c) \\ DTW_{SQ}(q[2...last], t_c[2...last]) \\ DTW_{SQ}(q, t_c[2...last]) \end{cases} \quad (2)$$

where $q[2...last]$ denotes the subsequence of q containing values from the 2nd to the last offset. To avoid pathological warping and reduce the quadratic computational cost, many research work [5] suggest to limit the warping length r such that $q[i]$ is matched with $t_c[j]$ if and only if $|i - j| \leq r$. This reduces the complexity of DTW from $O(m^2)$ to $O(mr)$.

To the best of our knowledge, the UCR Suite [5] is the state-of-the-art solution for arbitrary length subsequence search. Since exact distance computations are expensive, the UCR Suite applies a suite of lower-bounds to prune unpromising subsequences, before computing the exact distances for the remaining subsequences. Nevertheless, the subsequence search problem is still a computation intensive problem, especially for increasingly long time series nowadays. Even with the UCR Suite, the subsequence search on a trillion-scale time series would take 3.1 hours (under the Euclidean distance) or 34 hours (under Dynamic Time Warping) on a commodity PC [5].

To reduce the computation time of subsequence search, we propose effective lower-bounds based on the triangle inequality and Piecewise Aggregate Approximation (PAA). The proposed lower-bounds can be computed online and easily integrated into the UCR Suite. According to our experimental evaluations, the proposed methods can improve the performance of the UCR Suite by up to an order of magnitude.

The paper is organized as follows. Section 2 formally defines our problem and presents the state-of-the-art solution (i.e., the UCR Suite). We present our online lower-bounds in Section 3. The experimental study is given in Section 4. Finally, we conclude the paper in Section 5.

2. PRELIMINARIES

2.1 Problem Definition

In this work, we follow the suggestion of UCR Suite [5] that every subsequence must be Z-normalized in order to capture the similarity between the shapes of the sequences. Given a time series t, the Z-normalized value of $t[i]$ can be calculated as: $\hat{t}[i] = \frac{t[i] - \mu_t}{\sigma_t}$, where $t[i]$ is the i-th element of t, $\hat{t}[i]$ is the Z-normalized value of $t[i]$, and μ_t and σ_t are the mean and standard deviation of t, respectively.

PROBLEM 1 (SUBSEQUENCE SEARCH PROBLEM). *Given a time series t of length n, a query time series q of length m, and a distance function $dist(\cdot,\cdot)$, the subsequence search problem returns a length-m subsequence $t_c \in t$ such that $dist(\hat{q},\hat{t}_c) \leq dist(\hat{q},\hat{t}_{c'})$, $\forall\, t_{c'} \in t$ of length m.*

A naïve solution for the subsequence search (cf. Problem 1) is to calculate the distance $n-m+1$ times. The computation of the subsequence search may become prohibitive for long sequences. It should be noted that the search performance is closely related to the distance function $dist(\cdot,\cdot)$.

2.2 The State-of-the-Art: UCR Suite

In the following, we briefly introduce how the UCR Suite can boost ED-based and DTW-based subsequence search.

UCR-ED. For ED-based subsequence search, the UCR-ED [5] first employs the early abandoning technique. In general, it attempts to early terminate the distance computation when the accumulated distance is already larger than the best-so-far distance. To further improve the performance, the UCR-ED proposes *reducing the Z-normalization cost* and *prioritizing the accumulation order*.

UCR-DTW. For DTW-based subsequence search, the UCR-DTW [5] employs a filter-and-refinement framework. It evaluates the DTW distance for a subsequence only if it survives from three lower-bounds, i.e., $LB_{Kim\mathbf{FL}}$, LB^{EQ}_{Keogh} and LB^{EC}_{Keogh}. More specifically, the lower-bounds are applied in an order starting from quick-and-dirty one to slow-and-accurate one, as shown in Figure 2. We briefly introduce these lower bounds as follows.

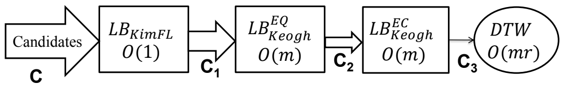


Figure 2: UCR-DTW framework

$\underline{LB_{Kim\mathbf{FL}}}$ is based on a fact that the **F**irst and the **L**ast offset must be matched in DTW.

$\underline{LB^{EQ}_{Keogh}}$ is derived from the distance between the candidate subsequence $\hat{t}_c$ and the envelop of $\hat{q}$. The envelop of $\hat{q}$ is based on the warping constraint r where $\hat{q}[i]$ can be matched with $\hat{t}_c[j]$ subject to $|j-i| \leq r$. The upper and lower envelop can be calculated as, $\hat{q}^u[i] = \max_{j=i-r}^{i+r} \hat{q}[j]$ and $\hat{q}^l[i] = \min_{j=i-r}^{i+r} \hat{q}[j]$, respectively. Accordingly, we have

$$LB^{EQ}_{Keogh}(\hat{q},\hat{t}_c) = \sqrt{\sum_{i=1}^{m} \begin{cases} (\hat{t}_c[i]-\hat{q}^u[i])^2 & \text{if } \hat{t}_c[i] > \hat{q}^u[i] \\ (\hat{t}_c[i]-\hat{q}^l[i])^2 & \text{if } \hat{t}_c[i] < \hat{q}^l[i] \\ 0 & \text{otherwise} \end{cases}} \quad (3)$$

$\underline{LB^{EC}_{Keogh}}$ is similar to LB^{EQ}_{Keogh} but the lower bound is derived from the distance between the query $\hat{q}$ and the envelop of $\hat{t}_c$. It should be noted that the optimization techniques in UCR-ED can be applied to compute LB^{EQ}_{Keogh} and LB^{EC}_{Keogh}.

3. LOWER-BOUND OPTIMIZATIONS

In this section, we propose two online lower-bounds, i.e., taking $O(1)$ and $O(\phi)$ time[1], to further improve the performance of the UCR Suite for both ED and DTW.

3.1 Fast ED-based Subsequence Search

The UCR Suite does not employ any lower-bound for the Euclidean distance (ED) [5], even though ED takes $O(m)$ time. In this section, we propose two lower bounds for ED, which can be computed in $O(1)$ and $O(\phi)$, respectively.

3.1.1 Lower-Bound by Triangle Inequality in $O(1)$ Time

As shown in Lemma 1, we can derive the Euclidean distance lower-bound for the running candidate t_c based on the exact distance of the last candidate (i.e., the consecutive subsequence t_{c-1}) by triangle inequality.

LEMMA 1 (LOWER-BOUND FOR ED). *For two consecutive candidate subsequences t_{c-1} and t_c in t, we have:*

$$ED(\hat{q},\hat{t}_{c-1}) - ED(\hat{t}_{c-1},\hat{t}_c) \leq ED(\hat{q},\hat{t}_c)$$

PROOF. It is trivial as triangle inequality holds for the Euclidean distance. □

We define $LB^{TRI}_{ED}(\hat{q},\hat{t}_c)$ as follows:

$$LB^{TRI}_{ED}(\hat{q},\hat{t}_c) = LB_{ED}(\hat{q},\hat{t}_{c-1}) - ED(\hat{t}_{c-1},\hat{t}_c), \quad (4)$$

where $LB_{ED}(\hat{q},\hat{t}_{c-1})$ is any lower bound which satisfies $LB_{ED}(\hat{q},\hat{t}_{c-1}) \leq ED(\hat{q},\hat{t}_{c-1})$.

With Lemma 1, we have $LB^{TRI}_{ED}(\hat{q},\hat{t}_c) \leq ED(\hat{q},\hat{t}_c)$. Suppose the $LB_{ED}(\hat{q},\hat{t}_{c-1})$ is known, we will elaborate how to compute it shortly. $LB^{TRI}_{ED}(\hat{q},\hat{t}_c)$ can be computed in $O(1)$ iff $ED(\hat{t}_{c-1},\hat{t}_c)$ can be calculated in constant time. To achieve this, it is sufficient to maintain five running sums: $S_1 = \sum_{i=1}^{m} t_{c-1}[i]$, $S_2 = \sum_{i=1}^{m} t_c[i]$, $S_3 = \sum_{i=1}^{m} t^2_{c-1}[i]$, $S_4 = \sum_{i=1}^{m} t^2_c[i]$, and $S_5 = \sum_{i=1}^{m} t_{c-1}[i]t_c[i]$, where these running sums can be maintained incrementally in $O(1)$ time. With these running sums, we can compute $ED(\hat{t}_{c-1},\hat{t}_c)$ in $O(1)$ as follows.

$$ED(\hat{t}_{c-1},\hat{t}_c) = \sqrt{2m(1-\rho(t_{c-1},t_c))} \quad (5)$$

where $\rho(t_{c-1},t_c)$ is the Pearson correlation between t_{c-1} and t_c.

$$\rho(t_{c-1},t_c) = \frac{mS_5 - S_1S_2}{\sqrt{mS_3-(S_1)^2}\sqrt{mS_4-(S_2)^2}} \quad (6)$$

3.1.2 Lower-Bound by PAA in $O(\phi)$ Time

The Piecewise Aggregate Approximation (PAA) [3] is a concise representation for time series. Given a normalized subsequence $\hat{t}_c$, its PAA representation is a ϕ-dimensional vector where the k-th element is defined as follows.

$$e_{\hat{t}_c}[k] = \frac{\phi}{\ell} \sum_{x=\frac{\ell}{\phi}\cdot k}^{\frac{\ell}{\phi}(k+1)-1} \hat{t}_c[x] \quad (7)$$

The distance, $LB^{PAA}_{ED}(\hat{q},\hat{t}_c)$, between the PAA representations of $\hat{q}$ and $\hat{t}_c$ is:

$$LB^{PAA}_{ED}(\hat{q},\hat{t}_c) = \sqrt{\frac{m}{\phi}\sum_{k=0}^{\phi-1}(e_{\hat{q}}[k]-e_{\hat{t}_c}[k])^2}$$

[1]The value of parameter ϕ is much smaller than m typically, i.e., $\phi \ll m$.

According to [3], $LB_{ED}^{PAA}(\hat{q}, \hat{t}_c) \leq ED(\hat{q}, \hat{t}_c)$. It can be calculated in $O(\phi)$ time with the PAA representations of $\hat{q}, \hat{t}_c$. Our remaining challenge is to compute the PAA of a subsequence efficiently, where a straightforward solution takes $O(m)$ time. To avoid $O(m)$ time cost, we transform Eq. 7 (i.e., expanding $\hat{t}_c[x]$) into the following equation.

$$e_{\hat{t}_c}[k] = \frac{\phi}{m} \sum_{x=\frac{m}{\phi}\cdot k}^{\frac{m}{\phi}(k+1)-1} \frac{t[c+x] - \mu(t_c)}{\sigma(t_c)} \tag{8}$$

Observe that $\mu(t_c)$, $\sigma(t_c)$, $\sum t[c+x]$ can be calculated in O(1) time by $\sum t_c[i]$, $\sum t_c^2[i]$ and $\sum t[c+x-1]$, respectively. Thus we can compute the PAA of $\hat{t}_c$ in $O(\phi)$ incrementally. This optimization is also utilized in [4].

3.1.3 *Putting Them Altogether*

Returning to the Equation 4, we require $LB_{ED}(\hat{q}, \hat{t}_{c-1}) \leq ED(\hat{q}, \hat{t}_{c-1})$. Thus, $LB_{ED}(\hat{q}, \hat{t}_{c-1})$ can be updated to $LB_{ED}^{TRI}(\hat{q}, \hat{t}_{c-1})$, $LB_{ED}^{PAA}(\hat{q}, \hat{t}_{c-1})$ and $ED(\hat{q}, \hat{t}_{c-1})$.

Algorithm 1 shows the pseudo code for ED-based subsequence search with our two novel lower bounds. We denote LB_{ED} as the lower bound for the current candidate t_c. We apply a cheap triangle inequality bound (cf. Line 5), before applying a slightly expensive PAA bound (cf. Line 7). If t_c cannot be pruned by all lower bound testings, then we compute the distance $ED(\hat{q}, \hat{t}_c)$ by calling UCR-ED. Furthermore, the value of LB_{ED} in the current iteration will be used to derive the bound in the next iteration.

Algorithm 1 ED-based subsequence search

```
    Alg ED-search(Query $q$, Sequence $t$, Dimensionality $\phi$)
1:  $os := -1, bsf := \infty, LB_{ED} := 0$
2:  for $c := 1$ to $n - m + 1$ do
3:      if $c > 1$ and $LB_{ED} > bsf$ then
4:          $LB_{ED} := LB_{ED} - ED(\hat{t}_{c-1}, \hat{t}_c)$              ▷ $O(1)$
5:      if $LB_{ED} < bsf$ then
6:          $LB_{ED} := LB_{ED}^{PAA}(\hat{q}, \hat{t}_c)$                   ▷ $O(\phi)$
7:          if $LB_{ED} < bsf$ then
8:              compute $ED(\hat{q}, \hat{t}_c)$ using UCR-ED               ▷ $O(m)$
9:              if $ED(\hat{q}, \hat{t}_c) < bsf$ then
10:                 $os := c, bsf := ED(\hat{q}, \hat{t}_c)$
11:             $LB_{ED} := ED(\hat{q}, \hat{t}_c)$
12: return $(os, bsf)$
```

We illustrate Algorithm 1 using an example in Figure 3. Suppose $bsf = 2$ and we can safely prune t_1 by computing its lower-bound, $LB_{ED}^{PAA}(\hat{q}, \hat{t}_1) = 10$. As $ED(\hat{t}_1, \hat{t}_2) = 4$ (i.e., using Equation. 5-6), thus we can derive $LB_{ED}(\hat{q}, \hat{t}_2) = (10 - 4) = 6$. Since $LB_{ED}(\hat{q}, \hat{t}_2)$ is larger than bsf, t_2 is safely pruned. Based on the triangle inequality, we have $ED(\hat{q}, \hat{t}_3) \geq LB_{ED}(\hat{q}, \hat{t}_2) - ED(\hat{t}_2, \hat{t}_3) = (LB_{ED}(\hat{q}, \hat{t}_1) - ED(\hat{t}_1, \hat{t}_2)) - ED(\hat{t}_2, \hat{t}_3)$. By computing $ED(\hat{t}_2, \hat{t}_3)$, we can derive $LB_{ED}(\hat{q}, \hat{t}_3) = (10 - 4) - 3 = 3 > bsf$, as a consequence, t_3 can also be pruned safely.

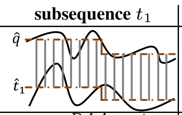
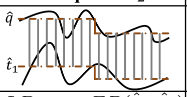
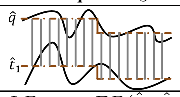
	subsequence t_1	subsequence t_2	subsequence t_3
Dist			
LB	$= LB_{ED}^{PAA}(\hat{q}, \hat{t}_1)$ $= 10$	$= LB_{ED} - ED(\hat{t}_1, \hat{t}_2)$ $= 10 - 4 = 6$	$= LB_{ED} - ED(\hat{t}_2, \hat{t}_3)$ $= 6 - 3 = 3$
Time	$O(\phi)$	$O(1)$	$O(1)$

Figure 3: Illustration of pruning techniques

3.2 Fast DTW-based Subsequence Search

The proposed lower-bounds in Section 3.1.1 can be adapted to $LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_c)$. In this section, we discuss two lower-bounds for LB_{Keogh}^{EQ} whose computation cost are $O(1)$ and $O(\phi)$, respectively.

3.2.1 *Lower-Bound by Triangle Inequality in $O(1)$ Time*

As shown in Lemma 2, we can derive the lower-bound of $LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_c)$ based on the $LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_{c-1})$ by triangle inequality.

LEMMA 2 (LOWER-BOUND FOR LB_{Keogh}^{EQ}). *Let t_{c-1} and t_c be consecutive candidates in t. We have:*

$$LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_{c-1}) - ED(\hat{t}_{c-1}, \hat{t}_c) \leq LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_c)$$

PROOF. For LB_{Keogh}^{EQ}, there exists a sequence a that is enclosed by the envelop sequences $\hat{q}^u$ and $\hat{q}^l$ (i.e., $\hat{q}^l[i] \leq a[i] \leq \hat{q}^u[i]$), such that $LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_c) = ED(a, \hat{t}_c)$. With Lemma 3, we have $LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_{c-1}) \leq ED(a, \hat{t}_{c-1})$.

$$LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_{c-1}) \leq ED(a, \hat{t}_{c-1})$$
$$\iff LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_{c-1}) - ED(\hat{t}_{c-1}, \hat{t}_c)$$
$$\leq ED(a, \hat{t}_{c-1}) - ED(\hat{t}_{c-1}, \hat{t}_c)$$

Applying Lemma 1: $ED(a, \hat{t}_{c-1}) - ED(\hat{t}_{c-1}, \hat{t}_c) \leq ED(a, \hat{t}_c)$

$$\iff LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_{c-1}) - ED(\hat{t}_{c-1}, \hat{t}_c) \leq ED(a, \hat{t}_c)$$
$$\iff LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_{c-1}) - ED(\hat{t}_{c-1}, \hat{t}_c) \leq LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_c)$$

□

LEMMA 3. *$LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_c) \leq ED(a, \hat{t}_c)$, where sequence a is enclosed by sequences $\hat{q}^u$ and $\hat{q}^l$, i.e., $\hat{q}^l[i] \leq a[i] \leq \hat{q}^u[i]$.*

PROOF. Since $\hat{q}^l[i] \leq a[i] \leq \hat{q}^u[i]$, for case (i) $\hat{t}_c[i] > \hat{q}^u[i]$, we have $(\hat{t}_c[i] - \hat{q}^u[i])^2 \leq (\hat{t}_c[i] - a[i])^2$. Similarly, for case (ii) $\hat{t}_c[i] < \hat{q}^l[i]$, we have $(\hat{t}_c[i] - \hat{q}^l[i])^2 \leq (\hat{t}_c[i] - a[i])^2$ as $a[i] \geq \hat{q}^l[i]$. Finally, $0 \leq (\hat{t}_c[i] - a[i])^2$ always hold for case (iii) $\hat{q}^l[i] \leq \hat{t}_c[i] \leq \hat{q}_u[i]$. □

Similarly, we define $LB_{DTW}^{TRI}(\hat{q}, \hat{t}_c)$ as follows:

$$LB_{DTW}^{TRI}(\hat{q}, \hat{t}_c) = LB_{DTW}(\hat{q}, \hat{t}_{c-1}) - ED(\hat{t}_{c-1}, \hat{t}_c), \tag{9}$$

where $LB_{DTW}(\hat{q}, \hat{t}_{c-1})$ is any lower bound with $LB_{DTW}(\hat{q}, \hat{t}_{c-1}) \leq LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_{c-1})$. With Lemma 2, we have $LB_{DTW}^{TRI}(\hat{q}, \hat{t}_c) \leq LB_{Keogh}^{EQ}(\hat{q}, \hat{t}_c)$.

3.2.2 *Lower-Bound by PAA in $O(\phi)$ Time*

In section 3.1.2, we propose a technique to construct the PAA representations for subsequences incrementally in $O(\phi)$ time. Given the PAA representations, we can derive $LB_{DTW}^{PAA}(\hat{q}, \hat{t}_c)$ as follows.

$$LB_{DTW}^{PAA}(\hat{q}, \hat{t}_c) = \sqrt{\frac{m}{\phi} \sum_{x=0}^{\phi-1} \begin{cases} (e_{\hat{t}_c}[x] - \hat{U}[x])^2 & e_{\hat{t}_c}[x] > \hat{U}[x] \\ (e_{\hat{t}_c}[x] - \hat{L}[x])^2 & e_{\hat{t}_c}[x] < \hat{L}[x] \\ 0 & \text{otherwise} \end{cases}}$$

where $\hat{U}$ and $\hat{L}$ are the PAA representation of the upper and lower envelopes of $\hat{q}$, respectively. In addition, its building cost can be neglected as it is only computed once at the beginning of the search.

According to [2], we have $LB^{PAA}_{DTW}(\hat{q}, \hat{t}_c) \leq LB^{EQ}_{Keogh}(\hat{q}, \hat{t}_c)$. Returning to Lemma 2, we require $LB_{DTW}(\hat{q}, \hat{t}_{c-1}) \leq LB^{EQ}_{Keogh}(\hat{q}, \hat{t}_{c-1})$. Thus, $LB_{DTW}(\hat{q}, \hat{t}_{c-1})$ can be set to $LB^{TRI}_{DTW}(\hat{q}, \hat{t}_{c-1})$, $LB^{PAA}_{DTW}(\hat{q}, \hat{t}_{c-1})$ or $LB^{EQ}_{Keogh}(\hat{q}, \hat{t}_{c-1})$.

3.2.3 Putting Them Altogether

Figure 4 illustrates the complete framework of our DTW-based subsequence search. We first evaluate $LB_{Kim\mathbf{FL}}$ as in the UCR Suite. We proceed to examine the lower-bound by triangle inequality if the candidates cannot be pruned by $LB_{Kim\mathbf{FL}}$. As a remark, $LB^{TRI}_{DTW}(\hat{q}, \hat{t}_c)$ is computed only when $LB_{DTW}(\hat{q}, \hat{t}_{c-1}) > bsf$. The surviving candidates are then evaluated by $LB^{PAA}_{DTW}(\hat{q}, \hat{t}_c)$, its cost is $O(\phi)$, before computing expensive LB^{EQ}_{Keogh} and LB^{EC}_{Keogh} (i.e., $O(m)$). Finally, we call $DTW(\hat{q}, \hat{t}_c)$ if $\hat{t}_c$ is not pruned by any above lower bounds.

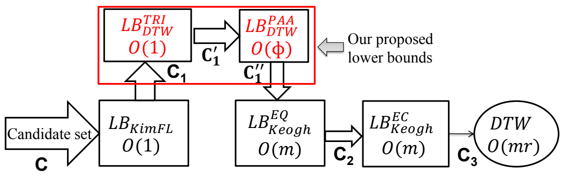


Figure 4: DTW-based subsequence search

4. EXPERIMENT

In this section, we compare our proposed techniques (denoted as FAST-ED and FAST-DTW), with the UCR Suite (denoted as UCR-ED and UCR-DTW).

Platform setting: All methods are implemented in C++. We evaluate the performance on a machine running 64-bit Windows 10 with a 3.16GHz Intel(R) Core(TM) Duo CPU E8500, 8 GB RAM.

Dataset: To evaluate the efficiency and scalability of our proposed techniques, we use the random walk model as in [4, 5] to generate both the query and the target time series. For each experimental setting, we run each method by 10 query sequences, and report the average execution time.

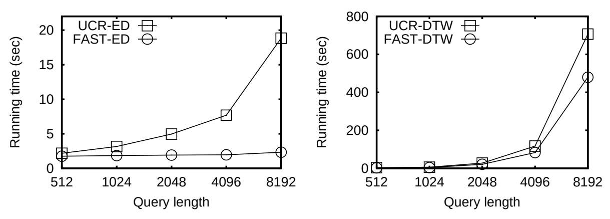
Result: We first test the performance of FAST-based subsequence search methods by varying ϕ from 8 to 40, where query length is 1024 and data length is 2 millions. The maximum standard deviation among these execution time is less than 3% in both FAST-ED and FAST-DTW. Thus, we choose $\phi = 24$ in the following experiments.

Table 1 compares the pruning ratio of our proposed lower-bounds with existing lower-bounds in the UCR Suite (in gray color). The UCR-ED does not provide any lower-bound function whereas our proposed lower-bounds can prune at least 99% of candidates. For example, it can prune 99.8% candidates when query length equals to 512 (i.e., among them, 14.6% candidates are pruned by LB^{PAA}_{ED} and 85.2% candidates are pruned by LB^{TRI}_{ED}). For subsequence search under DTW, our proposed lower-bounds are applied after LB_{KimFL} and before LB^{EQ}_{Keogh} (cf. Figure 4). Thus, the proposed lower-bounds can absorb most the pruning ability of LB^{EQ}_{Keogh}, but not other existing lower-bounds like $LB_{KimFL}, LB^{EC}_{Keogh}$. As shown in Table 1, our proposed lower-bounds can nearly approach the pruning ability of LB^{EQ}_{Keogh} by using less CPU time. Recall that our proposed bounds take $O(1)$ and $O(\phi)$ time, whereas LB^{EQ}_{Keogh} takes $O(m)$ time. Figure 5 shows

Table 1: Pruning ratios, on RW

Query Length	512	1024	2048	4096	8192
UCR-ED: N/A	-				
LB^{PAA}_{ED}	14.6%	11.8%	9.4%	7.1%	5.4%
LB^{TRI}_{ED}	85.2%	88.0%	90.3%	92.6%	94.2%
Total of proposed LBs	**99.8%**	**99.8%**	**99.7%**	**98.7%**	**99.6%**
UCR-DTW: LB^{EQ}_{Keogh}	44.0%	56.9%	80.9%	74.1%	95.0%
LB^{PAA}_{DTW}	12.0%	12.2%	12.1%	9.1%	7.9%
LB^{TRI}_{DTW}	30.8%	43.2%	67.2%	63.7%	86.4%
Total of proposed LBs	**42.8%**	**55.4%**	**79.3%**	**72.8%**	**94.1%**

the execution time of the FAST-based and the UCR-based subsequence search, by varying the query length, where data length is 2 million. The FAST-based search can be up to 11 and 3 times faster than the UCR-based search on ED and DTW respectively. Note that the performance gap between the FAST-based and the UCR-based widens when the query length increases.



(a) for ED (b) for DTW

Figure 5: Response time, on RW

5. CONCLUSION

In this paper, we propose two novel lower-bounds to speedup the subsequence search over time series data. These lower-bounds are based on the triangle inequality and PAA representations, respectively, and can be easily integrated with the UCR Suite to further improve its performance.

Acknowledgement

This work was supported by grant GRF 152043/15E from the Hong Kong RGC, grant MYRG105- FST13- GZG, MYRG2014-00106-FST, MYRG2015-00070-FST and MYRG2016-00182-FST from UMAC Research Committee, grant FDCT/116/2013/A3 and FDCT/007/2016/AFJ from Macau FDCT, grant NSFC 61502548 from National Natural Science Foundation of China. We thank Jiahao Zhang for helping with the experiments.

6. REFERENCES

[1] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *SIGMOD*, 1994.

[2] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.*, 7(3):358–386, 2005.

[3] E. J. Keogh, K. Chakrabarti, M. J. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowl. Inf. Syst.*, 3(3):263–286, 2001.

[4] Y. Li, L. H. U, M. L. Yiu, and Z. Gong. Quick-motif: An efficient and scalable framework for exact motif discovery. In *ICDE*, pages 579–590, 2015.

[5] T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. J. Keogh. Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *TKDD*, 7(3):10, 2013.